



**Modèles supersymétriques  
et mesures de précision à l'ILC**

SLAC Theory Group, MS 81  
2575 Sand Hill Road  
Menlo Park CA 94025 - USA

Julien Baglio  
Stage de recherche - PhyTEM 2ème année  
*sous la direction de JoAnne Hewett*

09 avril 2007 - 28 août 2007

## Remerciements

Je voudrais ici remercier tout d'abord le SLAC pour son accueil formidable et l'aide apportée tout au long des premiers jours, ainsi que pour m'avoir permis de participer au SLAC Summer Institute qui s'est déroulé du 30 juillet au 10 août 2007, et où j'aurais beaucoup appris en ce qui concerne la matière noire et la relation entre astrophysique et physique des particules autour de ce sujet. Merci à Robin Masset et Toni Campos pour l'installation au SLAC.

Je remercie aussi tout le groupe de physique théorique, à commencer par Michael Peskin le chef du groupe, pour son accueil, mais aussi Stan Brodsky pour sa bonne humeur et ses séminaires si particuliers, Roni Harnick avec qui j'ai partagé mon bureau un temps et qui a bien voulu répondre à certaines de mes questions, Guadalupe Makasyuk pour son aide technique sur tout ce qui concerne le réseau interne du SLAC, Lans Dickson, Daniel Maître pour les pauses cafés toujours bienvenue, et bien sûr les membres de mon équipe de travail : Jamie Gainer, le graduate avec qui j'ai partagé des moments intenses de physique, Tom Rizzo qui a guidé mes premiers pas en théorie au SLAC, et bien entendu JoAnne Hewett ma tutrice de stage qui a toujours été disponible et m'a encouragé depuis le début jusqu'à la fin.

## Abstract

Supersymmetry is a new symmetry enlarging the usual Lie algebra used in the Standard Model, in order to solve some of the many tiny problems found in our current theory of elementary particles and their interactions. Roughly, it exchanges bosons and fermions and thus groups them in what is called a supermultiplet. One of the key consequences of supersymmetry is the apparition of supersymmetric partners of the standard particles, which enlarge the Standard Model at least to what is named the MSSM : Minimal Supersymmetric Standard Model - its name meaning that it is the minimal extension in term of the number of new fields introduced.

Detailed studies have been conducted for hadronic colliders searches. The goal of my internship is to be part of a study on leptonic collider, in order to quantize precision measures and to study supersymmetric models which cannot be distinguished at LHC.

The first part of this report will deal with the basics of supersymmetry and explain why it was proposed nearly thirty years ago. Then I will focus on the study of charginos and neutralinos and I will end on the LHC inverse problem.

## Résumé

La supersymétrie est une nouvelle symétrie généralisant les algèbres de Lie utilisées dans la description du Modèle Standard et permettant de résoudre nombre de problèmes théoriques au sein de notre modèle actuel des particules élémentaires et de leurs interactions. En résumé, la supersymétrie échange bosons et fermions, ce qui en fait deux composantes d'un même objet physique que l'on appelle supermultiplet. Ainsi, une des principales conséquences de la supersymétrie est la prédiction de nouvelles particules dites supersymétriques partenaires de nos particules standards, ce qui élargit le Modèle Standard au MSSM, Modèle Standard Supersymétrique Minimal - son nom signifiant que c'est la plus petite extension possible en ce qui concerne le nombre de nouveaux champs introduits.

De nombreuses études ont été conduites pour étudier la supersymétrie auprès de collisionneurs hadroniques ; le but de mon stage est l'étude de la supersymétrie auprès d'un collisionneur leptonique, afin de quantifier des mesures de précision et d'étudier des modèles non discernables au LHC.

La première partie de mon rapport introduit les concepts de base de la supersymétrie et va expliquer quelles sont les raisons qui ont motivé sa création. Le rapport, dans un second temps, traite de l'étude des charginos et des neutralinos auprès de l'ILC, puis se finit dans un troisième temps avec l'étude du problème inverse du LHC.

# Table des matières

<b>1</b>	<b>Introduction à la supersymétrie</b>	<b>1</b>
1.1	Pourquoi la supersymétrie? . . . . .	1
1.2	Bases de la supersymétrie . . . . .	2
1.3	Superchamps et superespace . . . . .	4
1.3.1	Nombres de Grassmann et superespace . . . . .	4
1.3.2	Superchamp chiral et vecteur . . . . .	4
1.4	Brisure de symétrie . . . . .	6
1.4.1	Pourquoi briser . . . . .	6
1.4.2	Brisure souple . . . . .	7
1.4.3	Brisure spontanée avec termes F et D, mSugra . . . . .	7
<b>2</b>	<b>Extension minimale du Modèle Standard</b>	<b>8</b>
2.1	Un bref rappel du Modèle Standard . . . . .	8
2.2	Contenu en champ du MSSM . . . . .	10
2.2.1	R-parité et stabilité du proton . . . . .	10
2.2.2	Doublets de Higgs . . . . .	11
2.2.3	Panorama complet du MSSM . . . . .	11
<b>3</b>	<b>Étude du secteur de brisure de la supersymétrie</b>	<b>12</b>
3.1	Description des charginos et des neutralinos . . . . .	13
3.1.1	Des gauginos et higgsinos aux neutralinos et charginos . . . . .	13
3.1.2	Matrice de masse . . . . .	13
3.2	Composition du neutralino et production à l'ILC . . . . .	14
3.2.1	Production du neutralino . . . . .	14
3.2.2	Composition en higgsinos et gauginos . . . . .	15
3.2.3	Étude des modèles tests standards . . . . .	17
3.3	Production et désintégration des charginos . . . . .	19
3.3.1	Production et mélange du chargino . . . . .	19
3.3.2	Chaîne de désintégration . . . . .	19
3.3.3	Différence de masse neutralino-chargino . . . . .	21
<b>4</b>	<b>Le problème inverse du LHC</b>	<b>22</b>
4.1	Introduction à la problématique . . . . .	22
4.2	L'étude du groupe de Harvard . . . . .	23
4.2.1	Cadre de l'étude . . . . .	23
4.2.2	Paramètres étudiés . . . . .	24
4.2.3	Résultats : il existe un problème inverse du LHC! . . . . .	24
4.3	Étude sur l'ILC et création d'un fichier de contraintes expérimentales . . . . .	24
4.3.1	L'ILC et le problème inverse du LHC . . . . .	24
4.3.2	Contraintes astrophysiques et cosmologiques . . . . .	25
4.3.3	Contraintes liées aux mesures en accélérateur . . . . .	26
4.3.4	Problèmes rencontrés et modèles exclus . . . . .	27
4.3.5	La solution avec une interface Isajet-DarkSUSY couplée à Micromegas? . . . . .	27
	<b>Conclusion</b>	<b>28</b>
<b>A</b>	<b>Programme sur les bornes expérimentales</b>	<b>i</b>
<b>B</b>	<b>Interface Isajet/DarkSUSY et Isajet/Micromegas</b>	<b>iii</b>

# 1 Introduction à la supersymétrie

Mon stage s'est déroulé du 13 avril 2007 au 28 août 2007 au Stanford Linear Accelerator Center (SLAC), laboratoire d'État affilié à l'Université de Stanford et situé à Menlo Park en Californie. Il s'investit actuellement dans la collaboration BaBar, et s'est illustré notamment lors de la découverte de la sous-structure des nucléons dans les années 1970 ou bien lors de la découverte expérimentale du lepton tau en 1995.

J'ai travaillé au sein du groupe de physique théorique du SLAC, sous la direction de JoAnne Hewett, chercheur permanent du SLAC et professeur à l'Université de Stanford. Le but du stage consiste à étudier la phénoménologie de la supersymétrie auprès du futur grand collisionneur électron/antiélectron, encore en projet et appelé ILC pour International Linear Collider. Le LHC, collisionneur hadronique du CERN qui devrait démarrer en 2008, va nous ouvrir pour la première fois une fenêtre d'étude sur l'échelle du TeV en énergie et va nous permettre de tester un grand nombre de théories. Cependant, du fait de sa structure hadronique, il ne permet pas de faire des mesures de précision au contraire d'un collisionneur leptonique tel que l'ILC. C'est pourquoi des études théoriques en phénoménologie auprès de collisionneurs leptoniques sont essentielles.

Avant d'aborder le cœur de mon travail, il est nécessaire de poser les bases essentielles de la supersymétrie, ce qui est le but de cette première partie.

## 1.1 Pourquoi la supersymétrie ?

L'idée de la supersymétrie, qui consiste à relier bosons et fermions et à les considérer comme deux facettes d'un même objet, est née à peu près 30 ans plus tôt, dans le cadre de l'étude de la théorie (naissante) des cordes, sous l'impulsion de Julius Wess et Bruno Zumino. La structure mathématique quant à elle a été développée dans les années 60 par Gol'fand et Likhtman, qui n'ont pas appliqué à l'époque cette structure aux problèmes physiques en théorie des particules élémentaires.

La première version supersymétrique du Modèle Standard est née en 1981, grâce aux travaux de Howard Georgi et de Savas Dimopoulos : c'est le Modèle Standard Supersymétrique Minimal, ou MSSM. Il fut proposé pour trois raisons principales qui permettent d'expliquer l'attrait actuel de la supersymétrie :

1. Résoudre le problème de la hiérarchie. En effet, dans le cadre du Modèle Standard il existe des échelles d'énergie distinctes jouant un rôle capital : principalement l'échelle électrofaible, l'échelle de grande unification et l'échelle de Planck.

L'échelle électrofaible est l'échelle de brisure de la symétrie électrofaible. Dans la vie de tous les jours, les phénomènes électromagnétiques et faibles sont découplés, ce qui n'est pas le cas à haute énergie où ce sont deux aspects d'une même interaction appelée interaction électrofaible. Il existe donc une énergie caractéristique de cette unification, de l'ordre de  $10^2$  GeV.

De même, il existe une énergie caractéristique de grande unification entre l'interaction électrofaible et l'interaction forte, de l'ordre de  $10^{15}$  GeV, puis l'échelle de Planck où la gravitation doit rentrer en ligne de compte, de l'ordre de  $10^{19}$  GeV.

Comment rendre compte d'une telle différence d'ordre de grandeur entre ces trois énergies ? La supersymétrie offre une explication dans l'existence de particules supersymétriques offrant un pont entre l'énergie de grande unification et l'énergie électrofaible.

2. Aucun mécanisme au sein du Modèle Standard ne protège la masse du boson de Higgs qui est responsable de la brisure de symétrie électrofaible et qui permet de donner la masse aux particules du Modèle Standard. En effet, il existe des corrections dites radiatives à la masse du Higgs, et à moins d'un ajustement très fin des divers paramètres, ces corrections radiatives font exploser la valeur de cette masse.

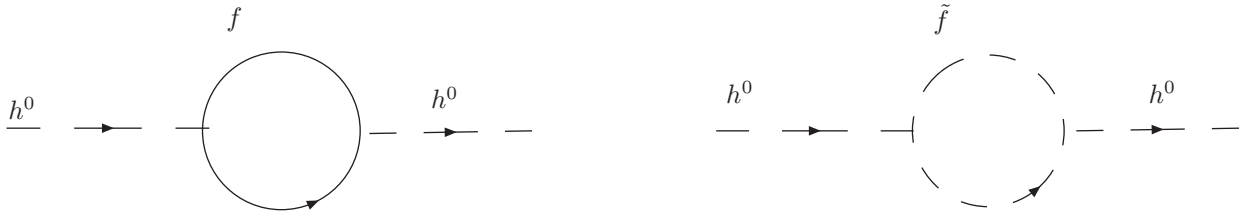


FIG. 1 – Corrections radiatives au Higgs  $h^0$ . A gauche, une contribution fermionique ; à droite une contribution bosonique, ici scalaire

Or les contributions des fermions et les contributions des bosons s'opposent, donc la supersymétrie permet par l'ajout de nouvelles particules partenaires d'annuler ces corrections radiatives et donc de stabiliser la masse du Higgs à l'échelle électrofaible.

3. La supersymétrie permet une réelle unification des constantes de couplages, comme le montre le schéma ci-dessous :

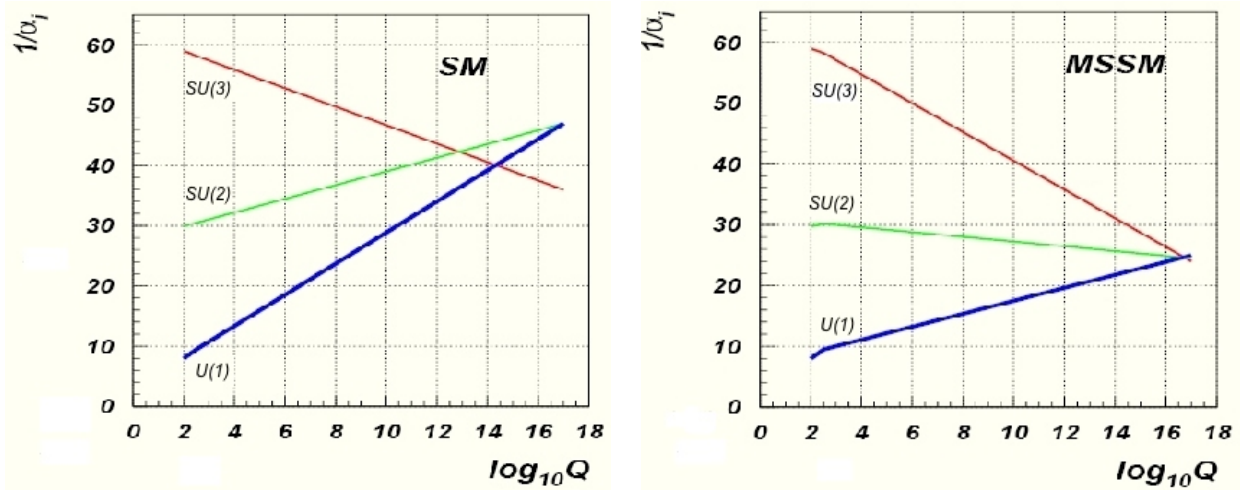


FIG. 2 – Unification des constantes de couplages  $\alpha_i$  fortes  $SU(3)$  et électrofaibles  $U(1)$  et  $SU(2)$ , en fonction de l'échelle d'énergie  $Q$ , dans le cadre du Modèle Standard à gauche, du MSSM à droite.

Depuis la supersymétrie a offert d'autres possibilités intéressantes dans d'autres domaines de la physique. Notamment dans le cadre du problème de la matière noire en astrophysique, la supersymétrie offre des pistes intéressantes car permettant de produire des particules massives stables et n'interagissant que très faiblement, appelées neutralinos.

## 1.2 Bases de la supersymétrie

Après avoir vu quelques-unes des raisons qui font de la supersymétrie une idée théorique intéressante, nous allons poser les bases de son formalisme. Nous nous appuyons sur [18] [15] et nous nous plaçons

dans une convention (+ - - -) pour la signature de l'espace de Minkowski - ce qui signifie que  $ds^2 = c^2 dt^2 - dx^2 - dy^2 - dz^2$ . De plus, la notation d'Einstein est utilisée dans ce rapport : quand deux indices identiques sont répétés, l'un en haut et l'autre en bas, ils sont implicitement sommés, c'est-à-dire  $a_i b^i \equiv \sum_i a_i b^i$ .

Nous postulons l'existence d'un opérateur  $Q$ , qui transforme un état fermionique en un état bosonique, et un état bosonique en un état fermionique. De ce fait,  $Q$  agissant sur des spineurs (qui décrivent les particules élémentaires fermioniques), doit être un opérateur spinoriel. Nous rappelons que les spineurs  $\psi^\alpha$  dans un espace 4D sont les solutions de l'équation de Dirac :

$$(i\gamma_{\alpha\beta}^\mu \partial_\mu - m)\psi^\beta = 0 \quad (1)$$

où les matrices  $\gamma_{\alpha\beta}^\mu$  sont les matrices de Dirac basées sur les matrices de Pauli, décrites ci-dessous dans la représentation de Weyl (dite aussi représentation chirale) :

$$\gamma^0 = \begin{pmatrix} 0 & Id \\ Id & 0 \end{pmatrix}, \quad \gamma^i = \begin{pmatrix} 0 & \sigma^i \\ -\sigma^i & 0 \end{pmatrix}$$

seuls les indices de Lorentz ont été représentés pour plus de clarté, les indices spinoriels  $\alpha\beta$  ayant été omis.

Les spineurs sont des objets mathématiques qui se transforment d'une certaine manière sous l'action du groupe de Lorentz décrivant les transformations spatio-temporelles, et qui n'est pas la même que la transformation des vecteurs. De plus les spineurs portent un spin demi-entier.

Le fait que l'opérateur  $Q$  soit un opérateur fermionique signifie que la supersymétrie est une symétrie de l'espace-temps. Or selon le théorème de Coleman-Mandula [18] il y a de fortes contraintes sur ce type d'opérateur. Celles-ci sont basées sur des hypothèses très générales sur la description de la matrice  $S$  utilisée pour décrire les interactions en théorie quantique des champs (et déjà rencontrée lors de l'étude de la diffusion quantique par un potentiel) :

- La matrice  $S$  est basée sur une théorie quantique relativiste des champs.
- Il existe un nombre fini de particules différentes associées à un état à une particule de masse donnée.
- Il y a un gap en énergie entre le vide et les états à une particule.

En utilisant le théorème de Coleman-Mandula, on obtient la forme suivante de l'algèbre engendrée par  $Q$  sur des spineurs à 2 composantes, que l'on appelle algèbre de Lie supersymétrique et généralisant les algèbres de Lie usuelles utilisées en physique quantique :

$$\begin{aligned} \{Q_\alpha, Q_\beta^\dagger\} &= 2\sigma_{\alpha\beta}^\mu P_\mu \\ \{Q_\alpha, Q_\beta\} &= \{Q_\alpha^\dagger, Q_\beta^\dagger\} = 0 \\ [P^m, Q_\alpha] &= [P^m, Q_\alpha^\dagger] = 0 \\ [P^m, P^n] &= 0 \end{aligned} \quad (2)$$

On voit donc que le plus important est l'introduction d'anticommutateurs, lié au fait que  $Q$  est fermionique. Afin d'appliquer la supersymétrie aux modèles existants et notamment au Modèle Standard, il faudra donc respecter l'algèbre supersymétrique (2), c'est-à-dire construire des objets physiques qui appartiennent aux représentations de ces algèbres (tout comme les vecteurs sont des éléments de représentations du groupe de Lorentz).

### 1.3 Superchamps et superspace

#### 1.3.1 Nombres de Grassmann et superspace

Nous avons vu précédemment que l'introduction d'anticommutateurs au sein d'une algèbre de commutateurs est une des caractéristiques de l'algèbre supersymétrique. Afin d'unifier la description des bosons et des fermions, on introduit le concept de superchamp agissant dans un superspace. Pour cela nous avons besoin de coordonnées décrivant l'anticommutation : les nombres de Grassmann.

Les nombres de Grassmann sont des éléments d'algèbres mathématiques qui anticommulent entre eux, et qui commutent avec les nombres ordinaires (les scalaires). Ainsi, pour les générateurs des algèbres en question, on a

$$\theta_i \theta_j = -\theta_j \theta_i, \quad \lambda \theta_i = \theta_i \lambda \quad (3)$$

Les  $\theta_i$  sont des nombres de Grassmann,  $\lambda$  est un scalaire. Un exemple de représentation des nombres de Grassmann est l'algèbre des matrices. Pour une algèbre de Grassmann à deux opérateurs on prendra ainsi

$$\theta_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \theta_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

Le superspace est donc l'espace de Minkowski habituel auquel on rajoute deux coordonnées de Grassmann  $(\theta, \bar{\theta})$ , qui sont des coordonnées à deux composantes spinorielles. Un superchamp est une fonction de la position au sein de ce superspace.

On peut développer le superchamp en ses coordonnées de Grassmann, et la propriété d'anticommutation nous donne par exemple pour une fonction sur des coordonnées à une composante spinorielle :

$$f(\theta, \bar{\theta}) = \lambda_0 + \lambda_1 \theta + \lambda_2 \bar{\theta} + \lambda_3 \bar{\theta} \theta + \lambda_4 \theta \bar{\theta}$$

car au-delà on a  $\bar{\theta}^2 = \theta^2 = 0$ . Les composantes  $\lambda_i$  seront les champs habituels fermioniques et bosoniques.

Dans ce rapport, afin de simplifier les notations, nous utilisons la notation de Weyl pour les spineurs à 4 composantes, c'est-à-dire que nous décomposons les spineurs à 4 composantes comme deux spineurs à deux composantes. Les nombres de Grassmann sont donc sur deux composantes. Dans la suite,  $\theta_a = (\theta_1, \theta_2)$ . On a  $\theta^a = \epsilon^{ab} \theta_b$  où  $\epsilon^{ab}$  est le tenseur antisymétrique :

$$\epsilon^{ab} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad \epsilon_{ab} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Ainsi  $\theta\theta = \theta^a \theta_a \neq 0$  mais chaque composante respecte le principe des nombres de Grassmann :  $\theta_1 \theta_1 = 0$  et  $\theta\theta\theta = 0$  ainsi que toute autre puissance supérieure.

#### 1.3.2 Superchamp chirale et vecteur

Nous avons maintenant toutes les cartes en main pour décrire les superchamps qui nous permettent de rendre le Modèle Standard supersymétrique. Nous introduisons les dérivées suivantes, qui sont une réalisation des opérateurs  $Q$  et  $Q^\dagger$  :

$$D_\alpha = \frac{\partial}{\partial \theta^\alpha} - i \sigma_{\alpha\beta}^m \bar{\theta}^\beta \frac{\partial}{\partial x^m}, \quad \bar{D}_\beta = -\frac{\partial}{\partial \bar{\theta}^\beta} + i \theta^\alpha \sigma_{\alpha\beta}^m \frac{\partial}{\partial x^m} \quad (4)$$

On appelle superchamp chirale un superchamp  $\Phi$  vérifiant  $\bar{D}\Phi = 0$ , superchamp vecteur un superchamp vérifiant  $\Phi^\dagger = \Phi$ . Soit le développement de  $\Phi$  en composantes suivant :



$$\begin{aligned} \Phi(x^\nu, \theta, \bar{\theta}) &= A(x^\nu) + i\theta\sigma^\mu\bar{\theta}\partial_\mu A(x^\nu) + \frac{1}{4}\theta\theta\bar{\theta}\bar{\theta}\square A(x^\nu) \\ &+ \sqrt{2}\theta\psi(x^\nu) - \frac{i}{\sqrt{2}}\theta\theta\sigma^\mu\partial_\mu\psi(x^\nu) + \theta\theta F(x^\nu) \end{aligned} \quad (5)$$

Ce développement s'obtient à partir d'un premier développement en  $x^\nu + i\theta\sigma^\mu\bar{\theta}$ . On vérifie bien que  $\bar{D}_\alpha\Phi = 0$  : on a ainsi décrit un superchamp chiral.

On peut écrire ainsi le lagrangien chiral libre comme  $\mathcal{L} = \frac{\partial\Phi^\dagger\Phi}{\partial\theta\theta\bar{\theta}\bar{\theta}}$  ce qui nous donne immédiatement

$$\mathcal{L} = i\partial_\mu\bar{\psi}\sigma^\mu\psi + A^*\square A + F^*F \quad (6)$$

On reconnaît une partie cinétique décrivant un fermion (spineur  $\psi$ ) et une partie cinétique décrivant un scalaire (le champ  $A$ ), tout deux sans masse. Pour vérifier ces assertions, on peut parler en dimension de masse. On rappelle que dans le système d'unités naturelles  $\hbar = c = 1$ , on a  $[E] = [m]$ , l'action  $S$  est de dimension 1, et  $[t] = [x] = [E]^{-1}$ . Il est donc commode de tout exprimer en terme de dimension de masse, ce qui signifie que temps et espace sont de dimension -1 (puisque de dimension  $[m]^{-1}$ ), l'énergie et la masse sont de dimension 1. On a donc aussi  $\theta$  de dimension  $-1/2$ . Ainsi un champ scalaire est de dimension 1 puisque le lagrangien est de dimension 4, un champ spinoriel est de dimension  $3/2$  selon l'équation (1).

Ici, on reconnaît donc par ce raisonnement aux dimensions que  $A$  est bien un champ scalaire,  $\psi$  est bien un champ spinoriel car  $\theta$  est de dimension  $-1/2$ . Il reste le champ  $F$  qui est de dimension 2 : cela signifie que le champ  $F$  n'est pas un champ physique mais un champ auxiliaire.

Avant de donner l'utilité du champ  $F$ , on mentionne la transformation supersymétrique sous laquelle le lagrangien (6) est invariant :

$$\begin{aligned} \delta A &= \sqrt{2}\zeta^\alpha\psi_\beta \\ \delta\psi &= i\sqrt{2}\sigma^m\bar{\zeta}\partial_m A + \sqrt{2}\zeta F \\ \delta F &= i\sqrt{2}\bar{\zeta}\sigma_m\partial_m\psi \end{aligned} \quad (7)$$

où  $\zeta$  est un spineur infinitésimal paramétrisant la transformation.

Quelle est l'utilité du champ auxiliaire  $F$  ? Il a été vu que sa dimension de masse en fait un champ non physique, ce qui signifie que l'on peut le supprimer sans changer la physique du problème. En effet, si l'on souhaite le lagrangien décrivant l'évolution d'un champ scalaire et d'un champ spinoriel libres, les deux premiers termes de (6) suffisent *a priori*.

Cependant, on souhaite garder l'invariance supersymétrique sous la transformation (7) même en dehors du cas classique, que l'on appelle sur couche de masse : en clair, on veut que le lagrangien soit bien invariant supersymétrique pour toutes les impulsions possibles en théorie quantique des champs, car il y a des corrections quantiques décrites par des boucles dans un diagramme de Feynman, qui nécessitent donc une intégration sur toutes les impulsions possibles. L'introduction du champ auxiliaire  $F$ , qui s'élimine physiquement à l'aide de son équation d'Euler-Lagrange triviale  $F = F^* = 0$ , permet de conserver l'invariance supersymétrique en dehors de la couche de masse, et donc d'être utilisé - heureusement ! - en théorie quantique des champs.

La description du superchamp vecteur suit la même démarche. On a aussi la présence d'un champ auxiliaire pour les mêmes raisons, et un développement selon les coordonnées de Grassmann. On mentionne rapidement le superchamp selon ses coordonnées :

$$V(x^\nu, \theta, \bar{\theta}) = -\theta\sigma^\mu\bar{\theta}v_m(x^\mu) + i\theta\theta\bar{\theta}\bar{\lambda}(x^\mu) - i\bar{\theta}\bar{\theta}\theta\lambda(x^\mu) + \frac{1}{2}\theta\theta\bar{\theta}\bar{\theta}D(x^\nu) \quad (8)$$

Le champ  $D$  est bien sûr réel, pour vérifier la condition  $V^\dagger = V$  ; il joue le rôle de champ auxiliaire. On reconnaît aisément la description d'un champ vectoriel réel  $v_m$  et d'un champ scalaire complexe  $\lambda$ . On regroupera donc par la suite les bosons de jauge et leur superpartenaires dans un tel superchamp vecteur, les leptons et les quarks ainsi que leurs superpartenaires dans des superchamps chiraux. Pour plus de précisions, on pourra se référer à [18], page 37.

## 1.4 Brisure de symétrie

### 1.4.1 Pourquoi briser

Nous avons vu ci-dessus comment généraliser les champs de la physique standard pour incorporer la supersymétrie. La supersymétrie associe bosons et fermions, donc si la supersymétrie est exacte il y a autant de degrés de liberté fermioniques que de degrés de liberté bosoniques au sein d'un supermultiplet. La démonstration est la suivante.

Soit un opérateur noté  $N_F$  tel que  $N_F|f\rangle = -1|f\rangle$  et  $N_F|b\rangle = +1|b\rangle$  où  $|b\rangle$  désigne un boson,  $|f\rangle$  désigne un fermion. On se place dans un espace de dimension finie et l'on travaille ici en composantes et non dans le superspace. On montre aisément que pour l'opérateur supersymétrique  $Q$  on a  $N_F Q = -Q N_F$ , en regardant l'action de ces deux produits d'opérateurs sur les deux classes de vecteurs indiquées ci-dessus. Ainsi on a

$$\text{Tr}(N_F\{Q_\alpha, Q_\beta^\dagger\}) = \text{Tr}(N_F Q_\alpha Q_\beta^\dagger) + \text{Tr}(Q_\alpha N_F Q_\beta^\dagger)$$

où le deuxième terme à droite de l'égalité provient de l'utilisation de  $\text{Tr}(AB) = \text{Tr}(BA)$ . Avec  $N_F Q = -Q N_F$  on aboutit alors à

$$\text{Tr}(N_F\{Q_\alpha, Q_\beta^\dagger\}) = 0$$

On utilise l'algèbre (2), ce qui donne

$$0 = \text{Tr}(N_F\{Q_\alpha, Q_\beta^\dagger\}) = 2\sigma_{\alpha\beta}^\mu \text{Tr}(N_F P_\mu) \quad (9)$$

Ceci est valable quel que soit  $P_\mu$ , donc  $\text{Tr}(N_F) = 0$  : il y a autant d'états fermioniques que bosoniques au sein de la représentation supersymétrique adoptée.

Il en résulte que l'on doit trouver dans la Nature autant de bosons que de fermions, aux propriétés identiques au spin près. Or si l'on souhaite, par exemple, trouver le superpartenaire de l'électron, on doit trouver un boson de spin 0 - en effet le spin 1 serait la marque d'un boson de jauge, or on parle ici de partenaire bosonique et non d'une nouvelle interaction - de même masse  $m = 511$  keV, de même charge électrique  $-e$ , etc... Il se trouve qu'aucun des bosons actuels ne fait l'affaire : la supersymétrie est donc brisée à notre échelle, afin que les superpartenaires de nos particules standard aient une masse élevée, ce qui explique que l'on n'a pas encore pu les observer.

Il s'agit donc de décrire le mécanisme de brisure de la supersymétrie. Or l'affaire est ici délicate, car on souhaite conserver toutes les raisons qui font que l'on a introduit la supersymétrie tout en fournissant un mécanisme cohérent de brisure de symétrie.

### 1.4.2 Brisure souple

L'idée naturelle qui vient à l'esprit pour la brisure de la supersymétrie est le mécanisme de brisure spontanée : la théorie est toujours supersymétrique, et donc en particulier le lagrangien, mais le vide de la théorie brise la supersymétrie :  $\langle 0|H_{SUSY}|0\rangle \neq 0$ . Ce mécanisme a été appliqué avec succès à la théorie électrofaible au sein du Modèle Standard, ce qui rend l'idée attirante pour briser la supersymétrie.

Cependant, nous souhaitons ici conserver les idées forces de la supersymétrie, c'est-à-dire corriger les effets radiatifs à la masse du Higgs, unifier à haute énergie les interactions, etc... Pour cela, les effets de brisure ne doivent pas perturber ceux que l'on souhaite conserver de la supersymétrie sous-jacente. On doit donc pouvoir séparer le lagrangien en  $\mathcal{L} = \mathcal{L}_{soft} + \mathcal{L}_{SUSY}$  où  $\mathcal{L}_{soft}$  contient tous les termes de brisure, c'est-à-dire les masses et couplages, mais de dimension positive de masse, et  $\mathcal{L}_{SUSY}$  est la partie conservant la supersymétrie.

Ainsi, on a effectué une brisure explicite de la supersymétrie et non une brisure spontanée, en rajoutant à la main les masses, et la brisure est dite souple car respectant des contraintes assurant les qualités de la supersymétrie même dans un contexte où elle est brisée. Cela se traduit dans la pratique par des termes de masse relativement contraints. Le Modèle Standard Supersymétrique Minimal (MSSM) est constitué de cette façon, et les termes de masses sont des paramètres libres du modèle.

Il faut pouvoir expliquer l'origine de cette brisure. Nous allons aborder dans le paragraphe suivant les diverses idées liées à l'utilisation du mécanisme de brisure spontanée. Il faut noter que ces mécanismes sont compliqués et ne font pas l'objet de l'étude de ce rapport : seuls les paramètres que l'on en déduit nous seront utiles.

### 1.4.3 Brisure spontanée avec termes F et D, mSugra

L'hamiltonien est relié aux générateurs supersymétriques. Les termes qui jouent un rôle clef dans la valeur de l'hamiltonien sont les champs auxiliaires. En effet, seuls ces champs peuvent éventuellement avoir une valeur non-nulle en opérant sur le vide, car nous souhaitons conserver la supersymétrie pour les champs chiraux et vecteurs introduits, et nous profitons du caractère non physique des champs auxiliaires. Ainsi, on veut

$$\langle 0| \sum_i F_i^* F_i + \frac{1}{2} \sum_a D_a D_a |0\rangle > 0 \quad (10)$$

où F et D sont les champs auxiliaires respectivement des superchamps chiraux, supermultiplets des fermions standards, et des superchamps vecteurs, supermultiplets des bosons de jauge standards. Ainsi, la condition (10) est respectée dès lors que les équations  $F_i = 0$  et  $D_a = 0$  ne sont pas respectées simultanément pour toutes les valeurs possibles des champs associés.

Il y a donc brisure par D, appelés D-term ou brisure Fayet-Iliopoulos, ou par F, appelés F-term ou brisure O'Raifeartaigh [7] [15].

Les termes de Fayet-Iliopoulos ne sont pas utilisés pour expliquer la brisure de la supersymétrie, car ils ne permettent pas de générer des masses aux superpartenaires de groupes de jauges non-abélien tel que le groupe de l'interaction forte. Les termes O'Raifeartaigh sont donc privilégiés, dont on donne un exemple ici d'un ensemble de trois superchamps chiraux soumis au superpotentiel  $W = -k\Phi_1 + m\Phi_2\Phi_3 + \frac{y}{2}\phi_1\phi_3^2$ . Ils engendrent un potentiel

$$V = F_1 F_1^* + F_2 F_2^* + F_3 F_3^* , \quad F_1 = k - \frac{y}{2}\phi_3^{*2}, F_2 = -m\phi_3^*, F_3 = -m\phi_2^* - y\phi_1^*\phi_3^*$$

$F_1 = 0$  et  $F_2 = 0$  ne sont pas compatibles, ce qui brise bien la symétrie. On peut montrer que ce potentiel donne alors, après minimisation, un spectre de masse différent entre bosons et fermions et correspond au final à  $V = k^2$ ,  $F_1 = k$ .

On voit que la brisure repose sur  $k$  qui est un paramètre mis à la main. Ceci subsiste dans le cadre du MSSM, ce qui signifie qu'il faut ajuster ce paramètre afin qu'il soit petit par rapport à l'échelle de Planck, pour respecter les conditions de brisure souple vues plus haut. Or il serait souhaitable d'expliquer cette valeur et son origine, en d'autres termes il faudrait que le champ auxiliaire  $F_1$  acquière une valeur moyenne dans le vide. Il faut donc étendre le MSSM car aucun champ auxiliaire en son sein ne peut développer une valeur moyenne non-nulle dans le vide, et il existe un certain nombre d'arguments qui impliquent que ces termes de brisure appartiennent à un secteur "caché" de la théorie, à haute énergie et dont les effets ne s'expriment à plus basse énergie que radiativement - le couplage entre les particules du MSSM et ce secteur caché est donc très faible.

Ceci est difficile à réaliser et fait l'objet de nombreuses études actuelles sur la supersymétrie. Il existe un certain nombre de mécanismes pour obtenir une médiation de l'interaction entre ces deux secteurs qui permette de communiquer la brisure de la supersymétrie à l'échelle du MSSM et donner naissance aux termes de brisure souple. L'un d'eux fut utilisé pour certaines études-test dans ce rapport : le mécanisme de brisure à l'échelle de Planck dont les effets sont transmis au MSSM par l'interaction gravitationnelle, ce qui constitue dans sa version minimale le mSUGRA pour minimal Supergravity.

Ceci repose sur une version locale quantifiée de la supersymétrie, qui s'apparente alors à la gravité. Les termes de masses et couplages souples sont du type  $m \simeq \frac{\langle F \rangle}{M_{Planck}}$  par analyse dimensionnelle, et du fait de l'échelle d'énergie où la brisure naît, il y a réduction à 5 paramètres principaux dont tous les masses et couplages souples du MSSM dépendent :

- $m_{1/2}$  qui est la masse commune des superpartenaires des bosons  $Z, W^+, W^-$  à l'échelle d'unification
- $m_0$  qui est la masse commune des bosons de Higgs du MSSM à l'échelle d'unification
- $A_0$  qui est un terme de couplage trilineaire à l'échelle d'unification
- $\tan \beta$  qui est un terme lié à la valeur moyenne dans le vide des champs de Higgs du MSSM, voir 2.2.2
- le signe du paramètre  $\mu$ , paramètre libre du MSSM

Ce sont ces paramètres qui sont utilisés dans la suite lors d'études basées sur [3] [4].

## 2 Extension minimale du Modèle Standard

Le but de cette deuxième partie qui se veut courte est de présenter la zoologie du MSSM. En effet, après avoir posé les bases de la supersymétrie et expliqué son architecture dans la première partie, il s'agit maintenant de décrire le contenu en particules de l'extension minimale du Modèle Standard. On rappelle à cet effet le Modèle Standard et son architecture, avant de présenter la R-parité et la stabilité du proton, puis enfin le contenu du MSSM

### 2.1 Un bref rappel du Modèle Standard

Le Modèle Standard est le modèle actuel qui décrit les particules élémentaires constitutrices de la matière et leurs interactions. La construction théorique du modèle est achevée à peu près vers 1973, les expériences effectuées afin de valider ce modèle étant encore en cours. Citons comme dernière découverte expérimentale de poids, la découverte du quark top en 1995, dernier élément de la famille des 6 quarks.

Les interactions sont décrites par le groupe de jauge  $SU(3)_C \otimes (SU(2)_L \otimes U(1)_Y)$ .  $SU(3)_C$  représente le groupe de l'interaction forte qui lie les quarks constitutifs des neutrons et protons,  $SU(2)_L \otimes U(1)_Y$  représente le groupe de l'interaction électrofaible, unification de l'interaction faible responsable des désintégrations radioactives et de l'interaction électromagnétique. Les particules subissant ces interactions appartiennent à des représentations de ces groupes, et possèdent des charges de couleur et électrique qui expriment l'intensité de la force forte et électromagnétique subies, ainsi qu'une hypercharge et un isospin qui rentrent en jeu pour décrire leurs interactions faibles. Une interaction est un échange de particules dites bosons de jauge, qui sont les médiateurs de l'interaction. Il y a ainsi 8 gluons qui véhiculent l'interaction forte et qui sont regroupés dans un octet de couleur, le photon qui véhicule l'interaction électromagnétique et les bosons  $W^+$ ,  $W^-$  et  $Z^0$  qui véhiculent l'interaction faible.

On regroupe les particules sous la forme de doublet chiraux d'isospin au regard de l'interaction faible, qui est une interaction chirale car ne s'exerçant que sur les fermions dites gauches : avec  $\gamma_5 = i\gamma_0\gamma_1\gamma_2\gamma_3$  on a  $\psi_L = \frac{1 + \gamma_5}{2}\psi$  qui est la partie gauche du spineur  $\psi$ . Les particules droites sont des singulets d'interaction faible, et ne subissent donc pas cette interaction. De plus on distingue les leptons qui sont des fermions légers, et les quarks qui se regroupent en états liés pour former les hadrons subdivisés en deux catégories : les baryons sont des états de trois quarks, les mésons sont des états quark-antiquark.

Nous avons évoqué le mécanisme de Higgs lors de la discussion à propos de la brisure de la supersymétrie. Ce mécanisme permet d'expliquer les masses des particules au sein du Modèle Standard. Il repose sur l'existence d'un doublet scalaire dit de Higgs, qui a une valeur moyenne non nulle dans le vide de la théorie. Cette valeur  $v$  donne naissance à un boson scalaire dit boson de Higgs, et donne la masse aux diverses particules du Modèle Standard par couplage entre ces dernières et le doublet de Higgs. Voici un tableau qui résume les propriétés des particules du Modèle Standard :

TAB. 1 – Propriétés des particules du Modèle Standard.  $M$  est la masse,  $Q$  la charge électrique en unité de la charge élémentaire  $e$ ,  $J$  le spin,  $(I^W, I_3^W)_{L,R}$  l'isospin faible selon la chiralité gauche (L) ou droite (R),  $(S, C, B, T)$  sont respectivement l'étrangeté, le charme, la beauté et la vérité. Enfin la case couleur signale si la particule subit l'interaction forte, dont l'intensité dépend de la charge de couleur (il y a trois charges : rouge, bleu et vert)

Type	Nom	$M$	$Q$	$J$	$(I^W, I_3^W)_{L,R}$	$(S, C, B, T)$	Couleur ?
LEPTONS	$e$	0, 511 MeV	-1	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, -\frac{1}{2})_L$	0	non
	$\nu_e$	< 2 eV	0	$\frac{1}{2}$	$(\frac{1}{2}, \frac{1}{2})_L$	0	non
	$\mu$	106 MeV	-1	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, -\frac{1}{2})_L$	0	non
	$\nu_\mu$	< 0.19 MeV, 90% CL	0	$\frac{1}{2}$	$(\frac{1}{2}, \frac{1}{2})_L$	0	non
	$\tau$	1, 777 GeV	-1	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, -\frac{1}{2})_L$	0	non
	$\nu_\tau$	< 18.2 MeV, 95 % CL	0	$\frac{1}{2}$	$(\frac{1}{2}, \frac{1}{2})_L$	0	non
QUARKS	$u$	1, 5 à 3 MeV	$\frac{2}{3}$	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, -\frac{1}{2})_L$	0	oui
	$d$	3 à 7 MeV	$-\frac{1}{3}$	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, \frac{1}{2})_L$	0	oui
	$c$	1, 25 GeV	$\frac{2}{3}$	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, -\frac{1}{2})_L$	(0, 1, 0, 0)	oui
	$s$	95 MeV	$-\frac{1}{3}$	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, \frac{1}{2})_L$	(-1, 0, 0, 0)	oui
	$t$	170.9 GeV	$\frac{2}{3}$	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, -\frac{1}{2})_L$	(0, 0, 0, 1)	oui
	$b$	4, 20 GeV	$-\frac{1}{3}$	$\frac{1}{2}$	$(0, 0)_R, (\frac{1}{2}, \frac{1}{2})_L$	(0, 0, -1, 0)	oui

Type	Nom	$M$	$Q$	$J$	$(I^W, I_3^W)_{L,R}$	$(S, C, B, T)$	Couleur ?
BOSONS DE JAUGE	$g$	0 eV	-1	1	(0, 0)	0	oui
	$\gamma$	0 eV	0	1	(0, 0)	0	non
	$W^\pm$	80.4 GeV	$\pm 1$	1	(1, $\pm 1$ )	0	non
	$Z^0$	91, 2 GeV	0	1	(1, 0)	0	non
HIGGS <sup>a</sup>	$h$	> 114, 4 GeV	0	0	(0, 0)	0	non

<sup>a</sup>La particule n'a pas encore été découverte expérimentalement. La limite sur sa masse provient de travaux auprès du LEP

Les masses des quarks sont des masses non mesurées directement, mais calculée dans un schéma de renormalisation appelé  $\overline{MS}$ . Toutes ces particules possèdent des antiparticules dont toutes les charges quantiques sont opposées. Toutes les valeurs sont extraites du Particle Data Group (PDG) [10].

## 2.2 Contenu en champ du MSSM

### 2.2.1 R-parité et stabilité du proton

Avant d'aborder la zoologie du MSSM, nous allons parler d'un point crucial de toute extension du Modèle Standard : la stabilité du proton.

En effet, toutes les expériences et observations à ce jour montrent que le proton est une particule stable, qui ne subit aucune désintégration spontanée. Ainsi, toute extension théorique du Modèle Standard doit respecter cette stabilité du proton et ne doit pas offrir un mécanisme la brisant aisément. En particulier, nous voulons que le MSSM respecte toujours cette stabilité.

Or rien ne nous interdit *a priori* de considérer ce diagramme de désintégration du proton au sein du MSSM :

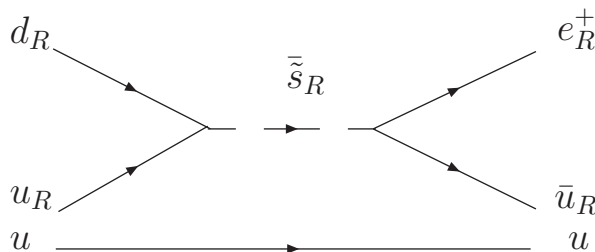


FIG. 3 – Diagramme de désintégration théorique du proton - à gauche le proton décomposé en ses quarks constitutifs, à droite le résultat de la désintégration avec un positron  $e^+$  et un pion  $\pi^0 = (u, \bar{u})$ , la particule scalaire au centre du diagramme étant un squark étrange, superpartenaire du quark étrange

C'est pour s'en prémunir que l'on a inventé le concept de R-parité. Ainsi, la R-parité est un nouveau nombre quantique multiplicatif associé aux particules, qui vaut +1 pour les particules standards et -1 pour les superpartenaires. Ce nombre quantique peut s'exprimer en fonction d'autres nombres quantiques :  $R = 3(B - L) + 2S$  où  $B$  est le nombre baryonique qui vaut 1 pour un baryon, -1 pour un anti-baryon, 0 pour les autres particules ;  $L$  est le nombre leptonique qui vaut 1 pour un lepton, -1 pour un anti-lepton et 0 pour les autres particules ;  $S$  enfin, qui est le nombre quantique de spin.

Ainsi dans la figure 3, la R-parité n'est pas conservée puisque par exemple au premier point d'interaction à gauche, on a une R-parité de +1 à gauche du point, -1 à droite. Une des conséquences fondamentales de la R-parité est la production en paires des particules supersymétriques, ainsi que la stabilité de la particule supersymétrique la plus légère, ce qui explique pourquoi cela en fait un candidat important pour expliquer la matière noire en astrophysique et en cosmologie.

### 2.2.2 Doublets de Higgs

On a vu qu'au sein du Modèle Standard, la brisure de symétrie électrofaible est réalisée à l'aide de valeur moyenne dans le vide (VEV) du champ du Higgs, ce qui donne aussi la masse aux particules élémentaires par interaction avec le doublet de Higgs.

Or dans le MSSM il y a besoin non pas seulement d'un doublet de Higgs, mais de deux doublets. Plusieurs raisons peuvent être données, l'une d'elle étant le fait que la supersymétrie relie bosons et fermions ; ainsi le partenaire supersymétrique d'un boson de Higgs doit se retrouver dans un doublet fermionique pour l'interaction faible, ce qui signifie donc deux bosons au départ. Une autre raison est liée à la génération des masses : on ne peut plus se servir uniquement d'un seul champ de Higgs et de son conjugué de charge, en raison de la construction même des superchamps qui sont chiraux. Ainsi pour les quarks l'un des deux doublets va générer les masses des quarks de type up, c'est-à-dire les quarks up, charme et top, tandis que le second doublet génère les masses des quarks de type down (down, étrange et beauté).

On a ainsi deux doublets de Higgs, l'un d'hypercharge +1 et l'autre d'hypercharge -1 :

$$\phi_1 = \begin{pmatrix} \phi_1^0 \\ \phi_1^- \end{pmatrix}, \quad \phi_2 = \begin{pmatrix} \phi_2^+ \\ \phi_2^0 \end{pmatrix}$$

Ils acquièrent alors une valeur moyenne non nulle dans le vide :

$$\langle \phi_1 \rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} v_1 \\ 0 \end{pmatrix}, \quad \langle \phi_2 \rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v_2 \end{pmatrix} \quad (11)$$

Soit  $v$  le VEV du champ de Higgs du Modèle Standard, on a alors  $v_1 = v \cos \beta$ ,  $v_2 = v \sin \beta$  et  $\tan \beta = \frac{v_2}{v_1}$  est un paramètre libre de la théorie. C'est ce paramètre qui va jouer un rôle dans les couplages et les masses et des fermions et bosons standard, et des partenaires supersymétriques.

On a 8 degrés de libertés réels puisque 4 champs complexes ; 3 de ces degrés de libertés servent à donner directement la masse aux bosons intermédiaires faibles, il reste donc 5 degrés de liberté qui sont les 5 bosons de Higgs suivants : le boson de Higgs neutre léger  $h^0$  assimilé au boson de Higgs standard le plus souvent, le boson de Higgs neutre lourd  $H^0$ , le boson de Higgs pseudo-scalaire neutre  $A^0$ , et les deux bosons de Higgs chargés  $H^\pm$ , et résultent de mélange entre les deux doublets. Voir [7] pour plus de détails, chapitre 10.

### 2.2.3 Panorama complet du MSSM

Avant de donner le panorama complet du MSSM, il est bon de signaler qu'il va y avoir un mélange de masse dans le secteur de brisure souple qui concerne les superpartenaires des bosons de jauge et des bosons de Higgs. Ce mélange donne naissance aux neutralinos et charginos, dont les signatures expérimentales ont fait l'objet de la première partie du stage et sont décrites dans la section 3.

D'avance on précise la nomenclature : un partenaire supersymétrique d'un boson a la même racine que la particule initiale, auquel on rajoute -ino, un partenaire supersymétrique d'un fermion a

le même nom que le fermion initial, auquel on rajoute un s au début.

Voici le tableau des particules superpartenaires :

TAB. 2 – Particules supersymétriques du MSSM.  $J$  est le spin des superpartenaires.

Type	Particule standard	Superpartenaire	symbole	$J$
LEPTONS	électron gauche, droit $e_L, e_R$	sélectron	$\tilde{e}_L, \tilde{e}_R$	0
	neutrino électronique $\nu_e$	sneutrino électronique	$\tilde{\nu}_e$	0
	muon gauche, droit $\mu_L, \mu_R$	smuon	$\tilde{\mu}_L, \tilde{\mu}_R$	0
	neutrino muonique $\nu_e$	sneutrino muonique	$\tilde{\nu}_\mu$	0
	tau gauche, droit $\tau_L, \tau_R$	stau 1, 2	$\tilde{\tau}_1, \tilde{\tau}_2$	0
	neutrino tauique $\nu_e$	sneutrino tauique	$\tilde{\nu}_\tau$	0
QUARKS	quark up gauche, droit $u_L, u_R$	squark up	$\tilde{u}_L, \tilde{u}_R$	0
	quark down gauche, droit $d_L, d_R$	squark down	$\tilde{d}_L, \tilde{d}_R$	0
	quark charme gauche, droit $c_L, c_R$	squark charme	$\tilde{c}_L, \tilde{c}_R$	0
	quark étrange gauche, droit $s_L, s_R$	squark étrange	$\tilde{s}_L, \tilde{s}_R$	0
	quark top gauche, droit $t_L, t_R$	squark top	$\tilde{b}_1, \tilde{b}_2$	0
	quark beauté gauche, droit $b_L, b_R$	squark beauté	$\tilde{t}_1, \tilde{t}_2$	0
BOSONS DE JAUGE	gluons $g$	gluinos	$\tilde{g}$	$\frac{1}{2}$
	photon $\gamma$	photino	$\tilde{\gamma}$	$\frac{1}{2}$
	bosons faibles chargés $W^\pm$	winos	$\tilde{W}^\pm$	$\frac{1}{2}$
	boson faible neutre $Z^0$	zino	$\tilde{Z}^0$	$\frac{1}{2}$
BOSONS DE HIGGS	boson de Higgs léger $h^0$	higgsino	$\tilde{h}_1^0$	$\frac{1}{2}$
	boson de Higgs lourd $H^0$	higgsino	$\tilde{h}_2^0$	$\frac{1}{2}$
	boson de Higgs pseudo-scalaire $A^0$			
	bosons de Higgs chargés $H^\pm$	higgsino	$\tilde{h}^\pm$	$\frac{1}{2}$

On donne ici quelques précisions. Tout d'abord, les nombres quantiques sont parfaitement identiques entre particules standards et particules supersymétriques, sauf en ce qui concerne le spin. C'est pourquoi ce tableau ne précise pas les nombres quantiques, il suffit de se référer au tableau 1 pour cela.

Ensuite, lorsqu'il y a un indice de chiralité pour les sfermions, c'est en référence à la chiralité du fermion correspondant, un boson n'ayant pas de chiralité.

De plus, les squarks de la 3<sup>e</sup> génération ont un mélange important entre les deux partenaires de chiralités définies, d'où la notation 1-2 plutôt que L-R.

Enfin, il y a quatre higgsinos et non 5, venant des deux doublets avant brisure, car ils serviront au mélange de masses. On introduit un paramètre de masse associé à ces higgsinos, qui est le paramètre  $\mu$  rencontré dans la section 1.4.3 .

### 3 Étude du secteur de brisure de la supersymétrie

Cette partie concerne la première partie du stage, du 9 avril à fin mai. C'est durant cette partie que j'ai pu me familiariser avec Pythia [9], un simulateur d'événements utilisé lors des études phénoménologiques auprès des collisionneurs. Le secteur de brisure souple de la symétrie, où se



déroule le mélange entre les higgsinos et les gauginos introduits lors de la deuxième partie, est le sujet principal de cette troisième partie.

Nous verrons d'abord comment décrire les charginos et les neutralinos et surtout de quels paramètres dépendent les masses et les coefficients de mélange. Puis nous verrons quelle est l'influence des diverses composantes - higgsino ou gaugino - sur les signatures de production de neutralinos auprès de l'ILC, et nous terminerons par une étude portant sur la désintégration du chargino léger ainsi que l'influence de la différence de masse chargino-neutralino sur le signal physique à l'ILC.

### 3.1 Description des charginos et des neutralinos

#### 3.1.1 Des gauginos et higgsinos aux neutralinos et charginos

Nous avons vu dans la deuxième partie que le MSSM contient 4 higgsinos dont deux chargés, et 4 gauginos dont deux chargés. Du fait de la brisure de la supersymétrie, nous pouvons leur associer des paramètres de masses, et du fait de la brisure de symétrie électrofaible ces 8 états se mélangent pour donner 8 états propres de masses appelés charginos et neutralinos.

Les neutralinos sont le résultat du mélange des higgsinos et gauginos neutres  $\tilde{h}_1$ ,  $\tilde{h}_2$ ,  $\tilde{\gamma}$  et  $\tilde{Z}^0$ . Les charginos sont le résultat du mélange des higgsinos et gaugino chargés  $\tilde{h}^\pm$  et  $\tilde{W}^\pm$ . On a donc 2 charginos et 4 neutralinos.

#### 3.1.2 Matrice de masse

##### Neutralinos

On débute avec la partie du lagrangien du MSSM qui nous intéresse, où l'on a des termes de masse ainsi que de couplages dit de Yukawa higgs-higgsino-gaugino :

$$\mathcal{L} = -\frac{g_2}{2}\tilde{W}^0(v_1\tilde{h}_1 - v_2\tilde{h}_2) + \frac{g_Y}{2}\tilde{\gamma}(v_1\tilde{h}_1 - v_2\tilde{h}_2) + \mu\tilde{h}_1\tilde{h}_2 - \frac{1}{2}M_1\tilde{\gamma}\tilde{\gamma} - \frac{1}{2}M_2\tilde{W}^0\tilde{W}^0 + \text{c.c.}$$

où c.c. désigne le complexe conjugué du premier terme. On retrouve donc les paramètres de masses  $M_1$  et  $M_2$  des gauginos, provenant de la brisure souple de supersymétrie, le paramètre de masse  $\mu$  des higgsinos, ainsi que les couplages  $g_2$  et  $g_Y$  des groupes de jauge  $SU(2)_L$  et  $U(1)_Y$ . Les constantes de couplages peuvent se relier à la masse du  $Z^0$  et à l'angle de Weinberg  $\theta_W$  qui caractérise le mélange des bosons intermédiaires.

On définit le vecteur  ${}^T\psi^0 = (\tilde{\gamma}, \tilde{Z}^0, \tilde{h}_1, \tilde{h}_2)$ , et l'on peut réécrire dans la base formé par ces quatre champs le lagrangien comme  $\mathcal{L} = -\frac{1}{2}{}^T\psi^0 M \psi^0 + \text{c.c.}$ , où  $M$  est la matrice de mélange définie comme

$$M = \begin{pmatrix} M_1 & 0 & -\cos\beta\sin\theta_W M_Z & \sin\beta\sin\theta_W M_Z \\ 0 & M_2 & \cos\beta\cos\theta_W M_Z & -\sin\beta\cos\theta_W M_Z \\ -\cos\beta\sin\theta_W M_Z & \cos\beta\cos\theta_W M_Z & 0 & -\mu \\ \sin\beta\sin\theta_W M_Z & -\sin\beta\cos\theta_W M_Z & -\mu & 0 \end{pmatrix} \quad (12)$$

Ainsi, les 4 neutralinos sont les états propres de cette matrice dans une base unitaire, on définit donc  $\tilde{\chi}_i = \sum_j X_{i,j}\psi_j^0$  où  $\psi_j^0$  sont les composantes du vecteur colonne  $\psi^0$ ,  $X$  est une matrice unitaire telle que

$$X^* M X^{-1} = \begin{pmatrix} m_{\tilde{\chi}_1} & 0 & 0 & 0 \\ 0 & m_{\tilde{\chi}_2} & 0 & 0 \\ 0 & 0 & m_{\tilde{\chi}_3} & 0 \\ 0 & 0 & 0 & m_{\tilde{\chi}_4} \end{pmatrix} \quad (13)$$

Il est d'usage d'ordonner les neutralinos du plus léger au plus lourd. Les angles de mélange et les masses ne dépendent donc que de quatre paramètres :  $M_1$ ,  $M_2$ ,  $\mu$  et  $\tan\beta$ . Ces paramètres seront choisis ici réels, et de signe positif sauf en ce qui concerne  $\mu$ . On n'étudiera donc pas d'effet de violation de CP dans cette matrice.

### Charginos

Le mécanisme est identique pour les charginos, qui sont un mélange entre les composantes chargées du secteur de brisure souple. On part du terme suivant dans le lagrangien du MSSM :

$$\mathcal{L} = -\frac{g_2}{\sqrt{2}} \left( v_1 \tilde{W}^+ \tilde{h}^- + v_2 \tilde{W}^- \tilde{h}^+ + \text{c.c.} \right) - \left( M_2 \tilde{W}^+ \tilde{W}^- + \mu \tilde{h}^+ \tilde{h}^- + \text{c.c.} \right) \quad (14)$$

On définit maintenant  ${}^T\psi^+ = (\tilde{W}^+, \tilde{h}^+)$  et  ${}^T\psi^- = (\tilde{W}^-, \tilde{h}^-)$  ce qui nous permet de réécrire le lagrangien [14] comme  $\mathcal{L} = {}^T\psi^- N \psi^+ + \text{c.c.}$ , avec

$$N = \begin{pmatrix} M_2 & \sqrt{2} \sin\beta M_W \\ \sqrt{2} \cos\beta M_W & \mu \end{pmatrix} \quad (15)$$

$N$  est la matrice de mélange des charginos, et l'on peut trouver deux matrices unitaires  $X$  et  $Y$  telles que

$$X^* N Y^{-1} = \begin{pmatrix} m_{\tilde{\chi}_1^\pm} & 0 \\ 0 & m_{\tilde{\chi}_2^\pm} \end{pmatrix}$$

ce qui nous permet de définir les charginos  $\tilde{\chi}_i^+ = \sum_j Y_{ij} \psi_j^+$  et  $\tilde{\chi}_i^- = \sum_j X_{ij} \psi_j^-$ . On constate que les masses sont les carrés des valeurs propres de  $N^\dagger N$ , ce qui permet de trouver leur expression analytique.

Ici les charginos ne dépendent aussi que de 4 paramètres : la masse des bosons  $W^\pm$   $M_W$ ,  $\tan\beta$  rapport des VEV des deux doublets de Higgs,  $\mu$  et  $M_2$ .

## 3.2 Composition du neutralino et production à l'ILC

On va maintenant étudier le type de signal que l'on peut obtenir dans un collisionneur leptonique. L'étude se place dans le cadre de l'ILC, le futur grand collisionneur linéaire international  $e^+e^-$ .

### 3.2.1 Production du neutralino

La production du neutralino auprès de l'ILC se fait à travers les trois diagrammes t, u et s suivants, à l'arbre :

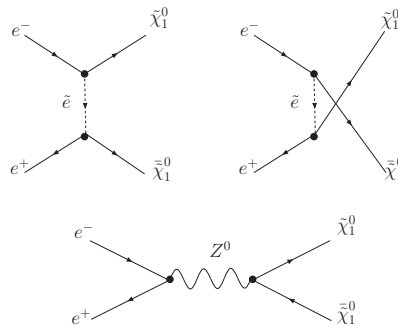


FIG. 4 – Diagrammes de production du neutralino au premier ordre dans une collision  $e^+e^-$

Notre étude dans le cadre de ce stage se concentre sur la distribution de photon radiatifs produits lors de la création de paires de neutralinos  $\tilde{\chi}_1^0 - \tilde{\chi}_1^0$  et  $\tilde{\chi}_1^0 - \tilde{\chi}_2^0$ . Ces photons viennent de diagrammes tels que

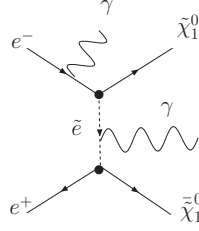


FIG. 5 – Production radiative de photons lors de la production de neutralinos à l’ILC

Le simulateur Pythia [9] est utilisé dans cette étude, en fixant l’énergie du centre de masse à  $\sqrt{s} = E_{cm} = 500$  GeV. Le nombre d’événements est proportionnel à la section efficace  $\sigma$ , calculée à partir des diagrammes de Feynman, ainsi qu’à la luminosité de l’accélérateur qui modélise son action sur les signaux de production. On a donc  $N = L \cdot \sigma$  où  $L$  est ce que l’on appelle la luminosité intégrée. Elle est prise ici égale à  $30 \text{ fb}^{-1}$ , ce qui est dans l’ordre de grandeur de la luminosité du LHC lors de son démarrage et très probablement de l’ILC.

### 3.2.2 Composition en higgsinos et gauginos

#### Signaux type dans la production des deux neutralinos les plus légers

Avant de s’intéresser à la composition du neutralino, voici le type de signal obtenu lors des simulations, faites sur 1 million d’événements :

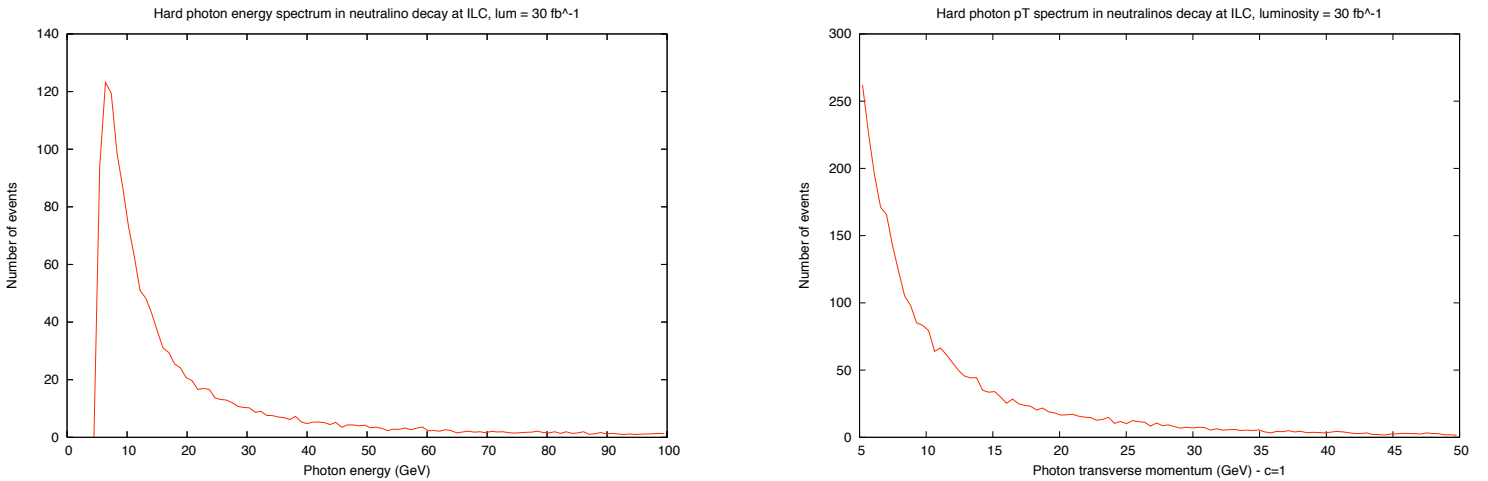


FIG. 6 – Spectre en énergie (à gauche) et en impulsion transverse (à droite) pour les photons radiatifs d’impulsion transverse supérieure à 5 GeV

On voit que l’énergie typique est de 10 GeV, avec une décroissance exponentielle. Ce type de signal est un moyen de détecter la production de neutralino dans une collision  $e^+e^-$ .

### Composition du neutralino léger et incidence sur les signaux

Le but de l'étude est d'étudier l'influence de la composition du neutralino le plus léger (LSP) sur les signaux obtenus. On va aussi chercher à les interpréter. Voici un exemple de signal :

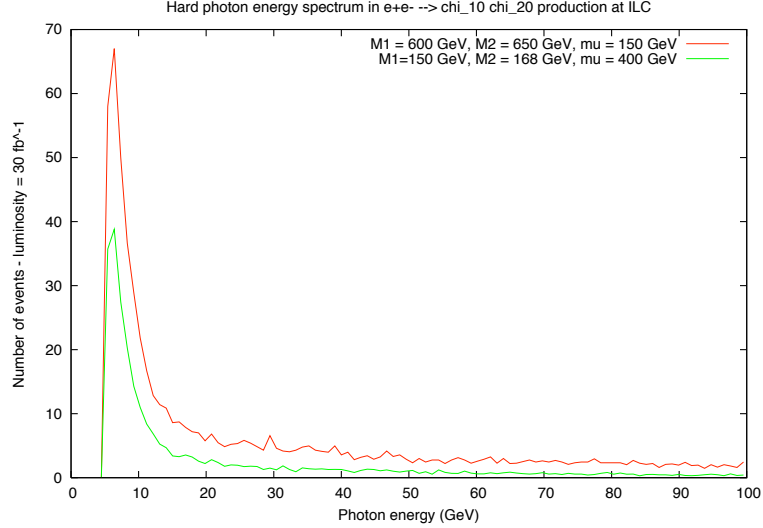


FIG. 7 – Spectre en énergie des photons radiatifs dans la production  $e^+e^- \rightarrow \tilde{\chi}_1^0 \tilde{\chi}_2^0$  : en rouge composition de type higgsino, en vert composition de type gaugino. On a  $\tan\beta = 10$

La courbe rouge est celle où le LSP est majoritairement du type higgsino, ayant  $M_{1,2} \gg \mu$ ; la courbe verte est au contraire typique d'un LSP de type gaugino. Quelle est l'importance de l'ordre des courbes ?

Pour répondre à cette question, étudions le cas deux deux LSP de même masse, mais de compositions très différentes comme ci-dessus :

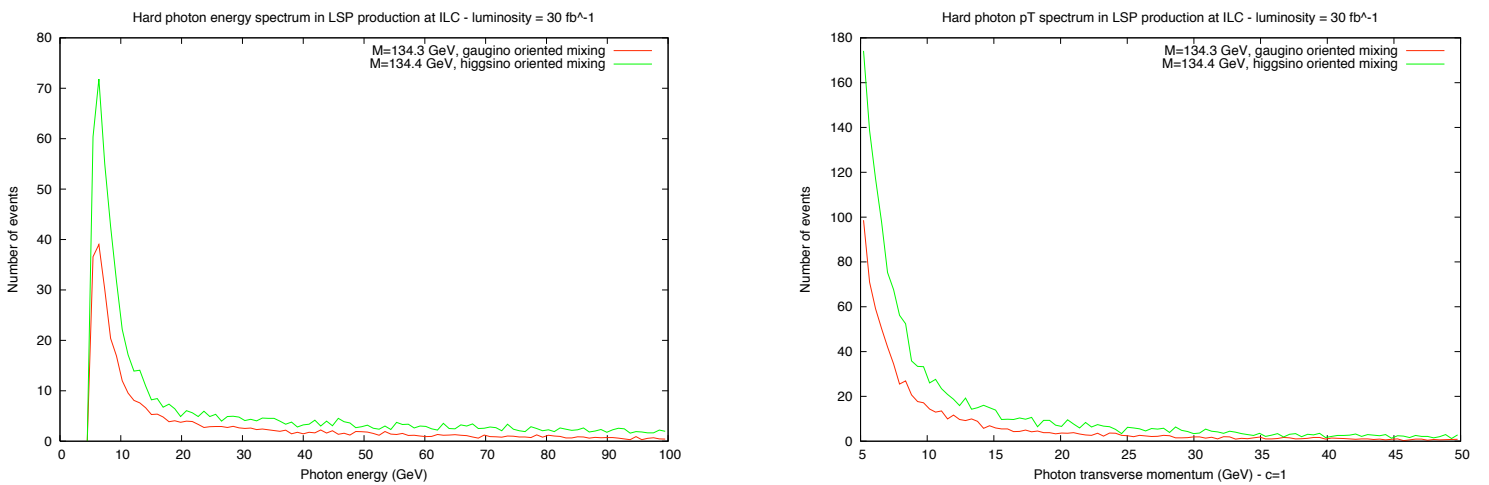


FIG. 8 – Spectre en énergie (à gauche) et en impulsion transverse (à droite) pour les photons radiatifs d'impulsion transverse supérieure à 5 GeV.  $M_{LSP} = 134.4$  GeV dans les deux simulations.

Dans le cas d'un LSP de type higgsino, le couplage électron/higgsino est de type couplage de Higgs : il est donc proportionnel à la masse de l'électron. Au contraire, si le neutralino est de type gaugino ce sont les couplages de jauge qui sont responsable de l'interaction, proportionnels à la masse du  $Z$ . Donc la section efficace est bien plus importante dans le cas d'un gaugino que d'un higgsino : l'ordre des courbes ne respecte pas cela !

### 3.2.3 Étude des modèles tests standards

Pour comprendre l'origine du problème, une étude test est entreprise avec des modèles tests standards. Ceux-ci sont extraits de [1], [3] et [4], et sont les scénarios LCC1, LCC2 et SPS1b, tous basés sur mSugra. Voici le spectre de masse des modèles utilisés et calculés avec Pythia :

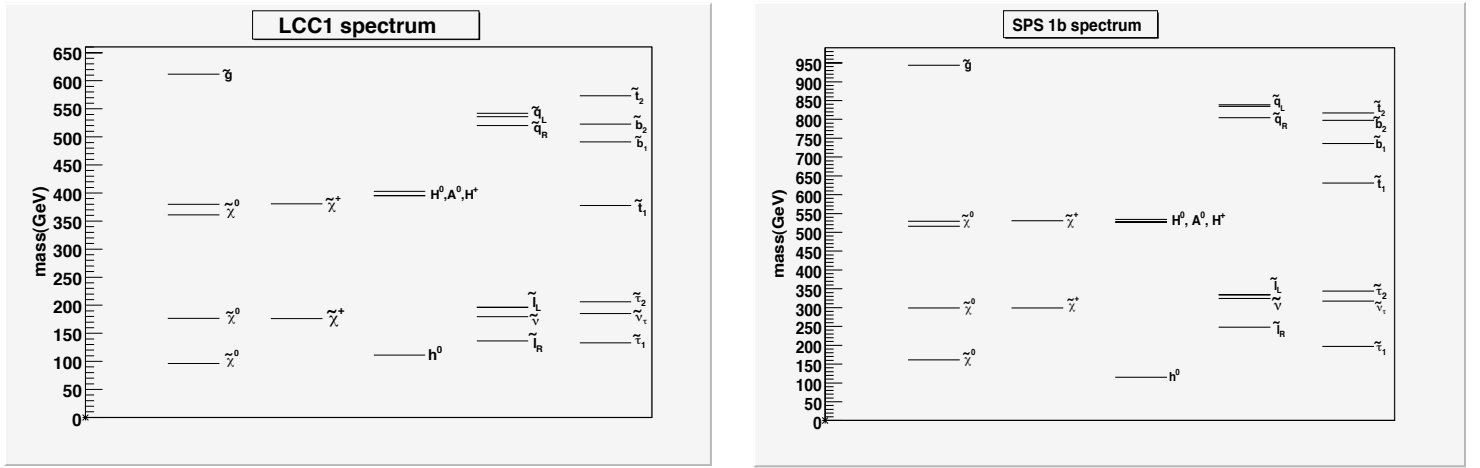


FIG. 9 – Spectre de masse des modèles LCC1 et SPS1b

Le spectre du modèle LCC2 est hors des limites présentées ci-dessus en ce qui concerne les squarks :

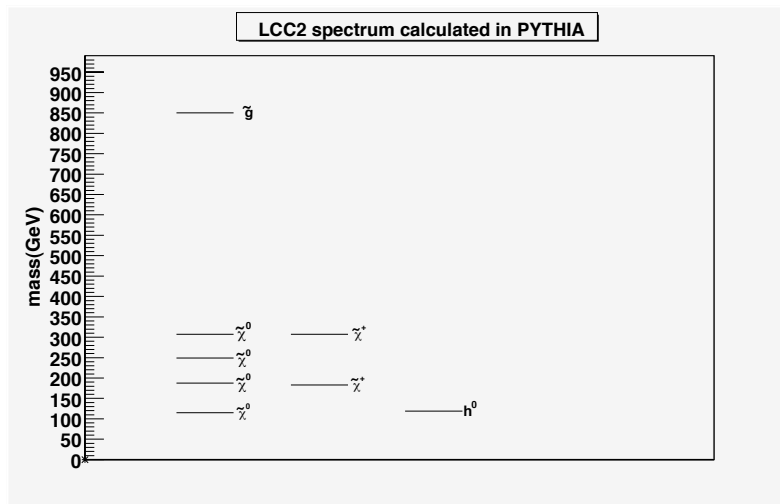


FIG. 10 – Secteur léger du modèle LCC2

On peut voir tout de suite la différence entre les modèles sur la section efficace  $\sigma(e^+e^- \rightarrow \tilde{\chi}_1^0 \tilde{\chi}_2^0)$  :

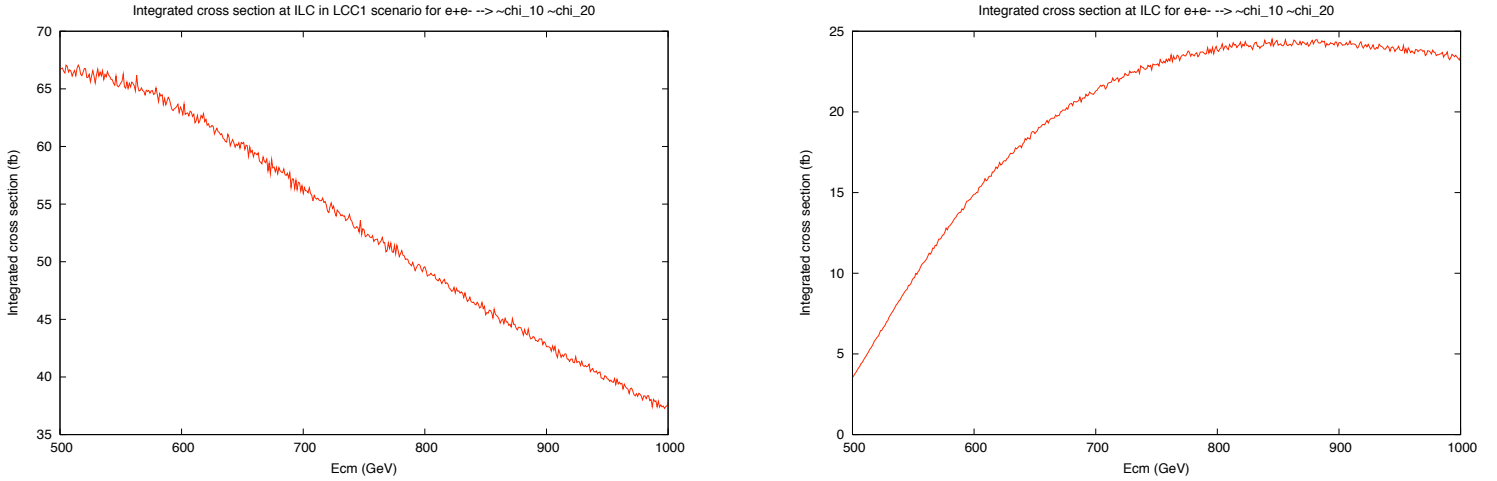


FIG. 11 – Évolution de la section efficace totale  $\sigma(e^+e^- \rightarrow \tilde{\chi}_1^0 \tilde{\chi}_2^0)$  en fonction de l'énergie  $E_{cm}$  : à gauche le modèle LCC1, à droite le modèle SPS1b

Or si l'on calcule avec Pythia les sections efficaces polarisées, on obtient

- $\sigma(e_R^+ e_L^- \rightarrow \tilde{\chi}_1^0 \tilde{\chi}_2^0) = 170$  fb dans le cadre LCC1 ; les données de (1) donnent 172 fb.
- $\sigma(e_R^+ e_L^- \rightarrow \tilde{\chi}_1^2 \tilde{\chi}_3^0) = 89$  fb dans le cadre LCC2 ; les données de [1] et [3] donnent plutôt 127.6 fb.
- $\sigma(e^+ e^- \rightarrow \tilde{\chi}_1^2 \tilde{\chi}_2^0) = 3.52$  fb (non polarisé). Cette valeur est en accord avec [12].

On voit donc que sur LCC2, notre étude avec Pythia est en désaccord avec les études déjà réalisées. Une étude attentive du spectre montre qu'en fait Pythia fournit des résultats erratiques à ce sujet, ce qui a influencé le signal obtenu : l'erreur venait donc d'un choix de paramètres où le simulateur choisi donne des résultats médiocres.

Par contre, sur un autre choix de paramètres, on retrouve l'ordre attendu entre les deux signaux :

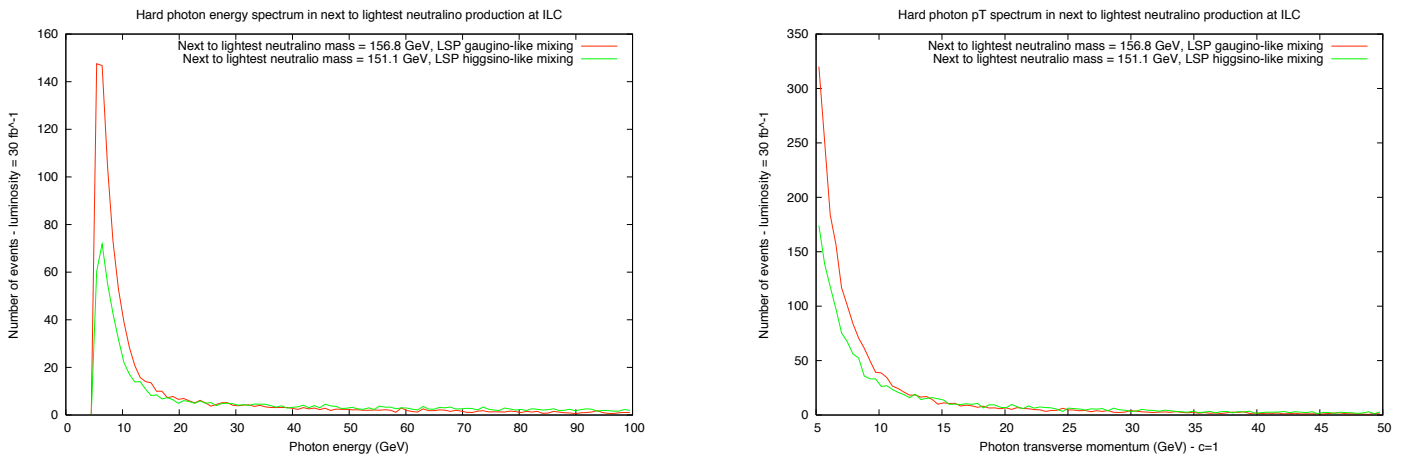


FIG. 12 – Spectre en énergie (à gauche) et en impulsion transverse (à droite) pour les photons radiatifs d'impulsion transverse supérieure à 5 GeV.  $M_{LSP} \simeq 152$  GeV dans les deux simulations.

Ainsi, cette étude a permis de mettre en évidence un des aspects fondamentaux des études phénoménologiques : croiser les données issues de simulateurs d'événements différents. On a pu mettre en évidence un autre ensemble de paramètres où le simulateur Pythia est à proscrire. Ces erreurs peuvent provenir d'instabilités numériques, voire de bugs internes.

### 3.3 Production et désintégration des charginos

Après avoir étudié les neutralinos, cette section est dévolue à l'étude des charginos qui sont les particules chargées du secteur de brisure souple de la supersymétrie. Après une présentation de sa production et de sa désintégration, nous verrons l'influence sur les signaux en collisionneurs de la proximité en masse des neutralinos et charginos les plus légers, abordée notamment dans [2] [12].

#### 3.3.1 Production et mélange du chargino

Le chargino est produit après collision d'un électron et d'un positron par échange de photon ou de boson  $Z^0$  dans les diagrammes s, ou par échange d'un sneutrino électronique dans les diagrammes t :

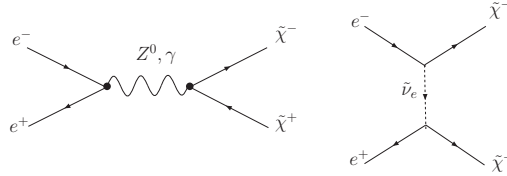


FIG. 13 – Diagrammes de production du chargino au premier ordre dans une collision  $e^+e^-$

La prédominance de l'un des diagrammes sur l'autre va dépendre de la composition du chargino : si le chargino est plutôt du type higgsino, c'est le diagramme t qui va prédominer, si le chargino est de type gaugino ce sera le diagramme s qui sera prédominant.

#### 3.3.2 Chaîne de désintégration

Le chargino est plus lourd que le LSP : il peut donc se désintégrer, et l'on peut détecter cette désintégration dans un collisionneur ce qui en fait un bon signal supersymétrique pour la détection expérimentale. Voici le diagramme de Feynman générique de la désintégration du chargino  $\tilde{\chi}_1^+$  :

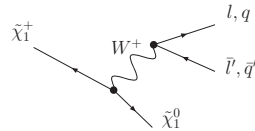


FIG. 14 – Désintégration du chargino en neutralino  $\tilde{\chi}_1^0$  et boson  $W^+$  qui lui-même se désintègre en paire quarks/antiquarks ou neutrino/antilepton

Le neutralino est invisible : il va emporter de l'énergie. Ce signal, associé à la production de jets+leptons est caractéristique de la supersymétrie et en fait un signal très étudié.

Voici une étude du signal obtenu sur les photons radiatifs dans la production du chargino :

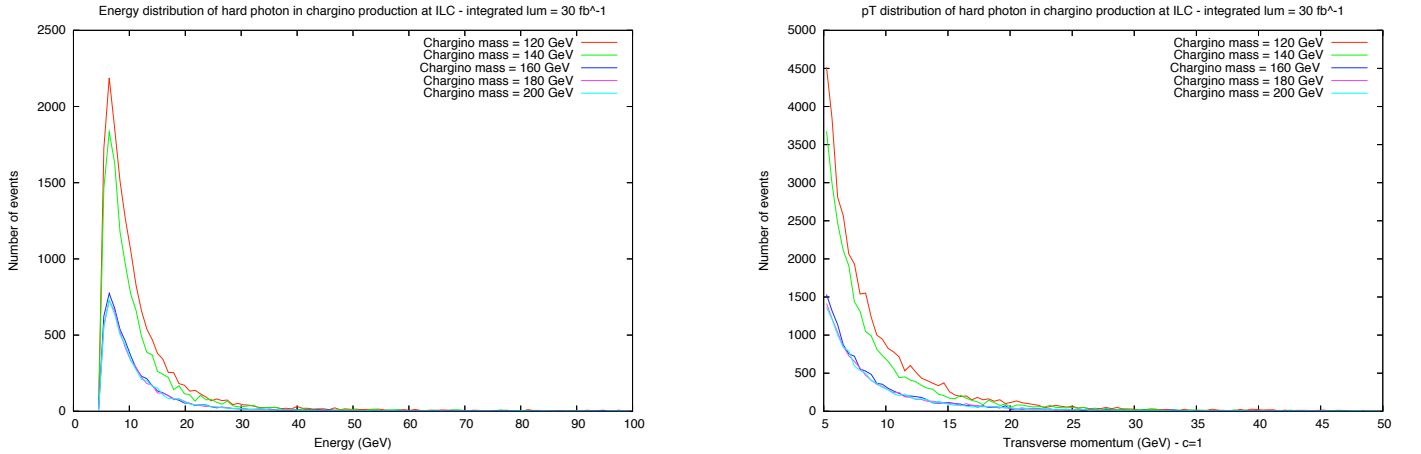


FIG. 15 – Spectre en énergie (à gauche) et en impulsion transverse (à droite) pour les photons radiatifs d’impulsion transverse supérieure à 5 GeV dans la production du chargino. L’étude est réalisée pour différentes masses.

L’ordre des courbes s’explique naturellement par une dépendance d’avec la masse : en effet plus la masse du chargino est importante, plus il est difficile de le produire à énergie fixée dans le centre de masse, car l’on réduit l’espace des phases sur lequel on intègre dans le calcul de la section efficace. Au contraire, plus la différence entre l’énergie de masse et l’énergie du faisceau est grande, plus le signal va être important.

Avant de passer à l’étude de la dépendance des signatures expérimentales avec  $\Delta m = m_{\tilde{\chi}_1^+} - m_{\tilde{\chi}_1^0}$ , voici un graphe de simulation sur les charginos qui compare un même signal tel qu’il est produit au LHC à 14 TeV, et à l’ILC à 500 GeV :

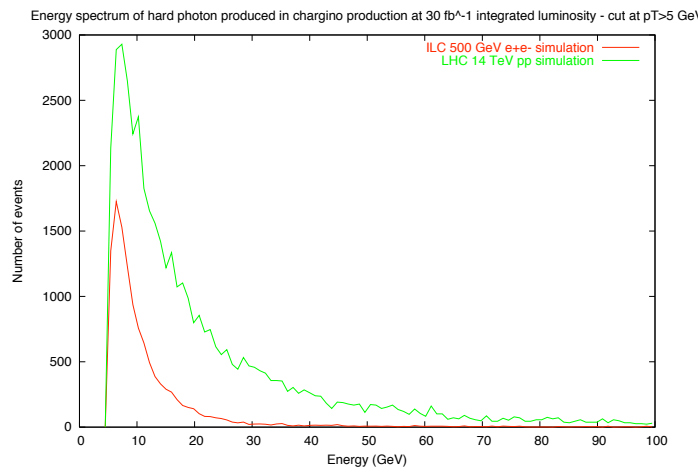


FIG. 16 – Comparaison entre le signal obtenu au LHC en collision pp à 14 TeV et le signal obtenu à l’ILC en collision  $e^+e^-$  à 0.5 TeV

On voit donc tout l’intérêt de l’ILC : le signal est plus piqué et permet donc des mesures précises,



nécessaires pour distinguer des modèles comme on le verra en 4e partie.

### 3.3.3 Différence de masse neutralino-chargino

Le processus de désintégration du chargino dépend très fortement de la masse du neutralino léger qui en résulte. Ceci motive une étude de la dépendance des signatures expérimentales d'avec la différence de masse  $\Delta m = m_{\tilde{\chi}_1^+} - m_{\tilde{\chi}_1^0}$  dans le régime des faibles différences de masses  $\Delta m < 10$  GeV. Pour que le signal soit utilisable dans le cadre de la chaîne passant par le boson  $W$  nous ne prendrons pas en compte les simulation des signaux tels que  $\Delta m < 0,5 \text{ GeV}$ .

Voici le spectre en masse typique des modèles utilisés pour cette étude, ce spectre ayant été vérifié par d'autres générateurs de spectres tels qu'ISAJET :

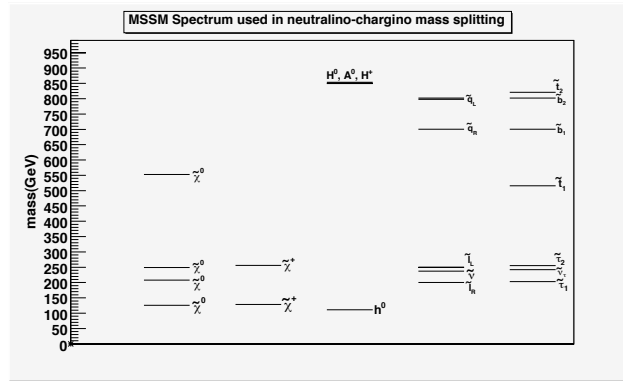


FIG. 17 – Spectre typique utilisé dans l'étude de  $\Delta m$ . On a bien une quasi-dégénérescence entre  $\tilde{\chi}_1^\pm$  et  $\tilde{\chi}_1^0$

Le seul paramètre sur lequel on va jouer est  $\mu$  car c'est ce paramètre qui est commun aux charginos et neutralinos, tous les autres paramètres restant inchangés. Voici donc le résultat de trois simulations pour des différences de masses de 1,22 GeV, 2,52 GeV et 4,65 GeV :

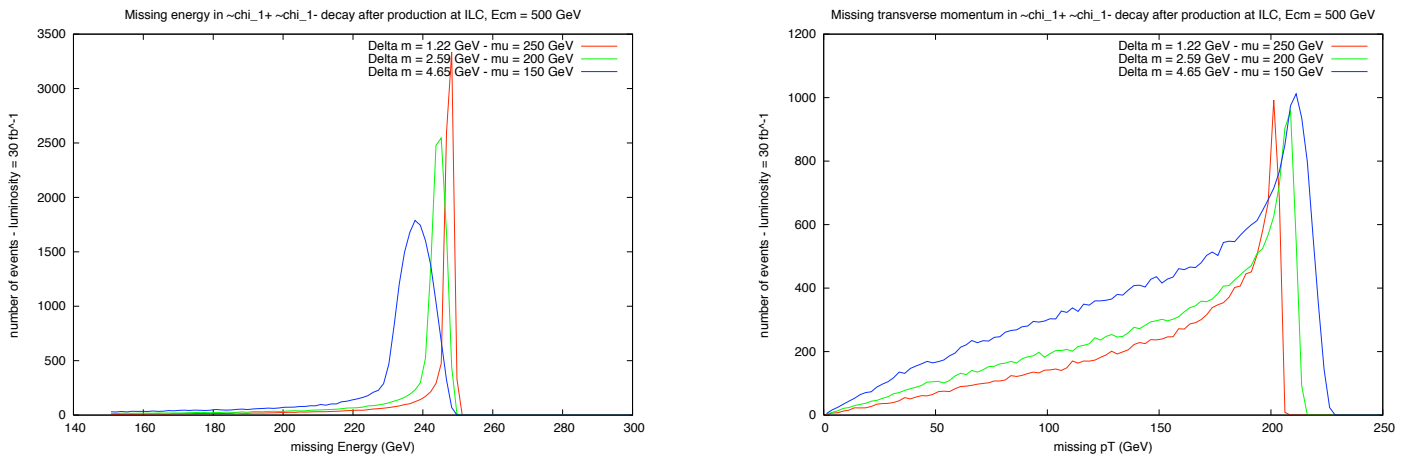


FIG. 18 – Spectre en énergie (à gauche) et en impulsion transverse (à droite) pour l'énergie et l'impulsion transverse manquantes dans la désintégration des charginos produits à l'ILC,  $E_{cm} = 500$  GeV et luminosité intégrée de  $30 \text{ fb}^{-1}$

Pour expliquer l'allure des courbes, il suffit de se souvenir que cinématiquement, on a nécessairement  $\cancel{E} = E(\tilde{\chi}_1^0) \geq m_{\tilde{\chi}_1^0}$  : ainsi la probabilité d'avoir un signal avec beaucoup d'énergie cinétique emportée par les autres produits de la désintégration est plus réduite si  $\Delta m$  est petit car il faut déjà enlever l'énergie de masse du neutralino. Les simulations se faisant sur un grand nombre d'événements (ici 1 million), cet effet s'observe statistiquement et explique que plus  $\Delta m$  est grand, plus l'énergie disponible autre que l'énergie de masse est grande et réduit d'autant plus le signal sur l'énergie manquante.

Au passage, remarquons que le pic de la distribution est à chaque fois localisé sur la masse du neutralino : c'est donc un signal qui permet de mesurer ladite masse.

Afin de voir s'il y avait une régularité dans cette dépendance entre énergie manquante et  $\Delta m$ , une étude de l'évolution des maxima ajustés de chaque distribution, en fonction de  $\Delta m$ , a été réalisée. Voici le résultat obtenu, ainsi que l'ajustement réalisé dans la gamme 1 – 5 GeV en différence de masse :

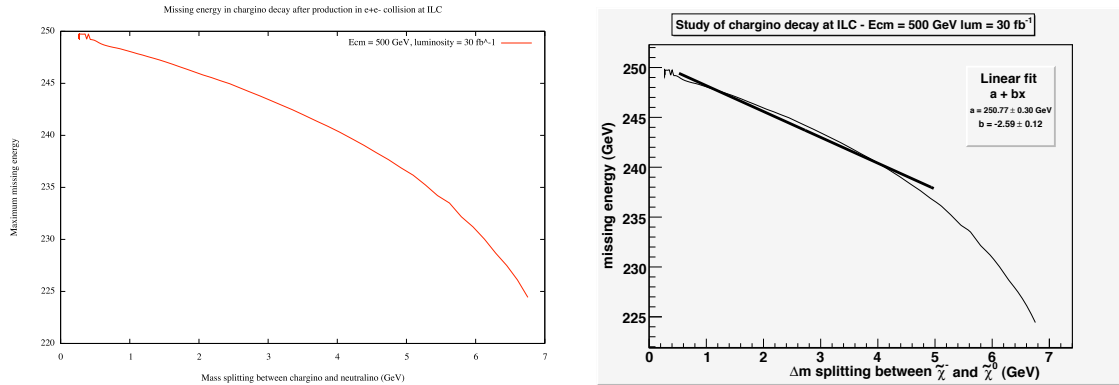


FIG. 19 – Énergie manquante dans la désintégration du chargino à l'ILC en fonction de  $\Delta m$ , ainsi que son ajustement

On trouve ainsi une dépendance linéaire :  $\cancel{E} = 250 - 2,59\Delta m$  (GeV) en unité  $c = 1$ .

## 4 Le problème inverse du LHC

Après avoir étudié les signaux intéressants du secteur de brisure souple de la supersymétrie dans le cadre de l'ILC, notamment dans le but de faire des mesures précises de masses, mon stage s'est dirigé vers un projet que j'ai construit avec mon équipe, qui s'inscrit dans un projet plus large qui est relié à ce que l'on appelle le problème inverse du LHC. Après une introduction expliquant la problématique ainsi qu'un résumé de l'étude sur laquelle se base notre travail, le coeur du projet sera abordé en détail.

### 4.1 Introduction à la problématique

Le problème inverse du LHC se situe dans le contexte du dialogue entre expérience et théorie. En effet, lorsque une étude théorique d'un modèle est conduite, on prend des points dans ce que l'on nomme l'espace des paramètres, c'est-à-dire que l'on utilise un ensemble de paramètres déterminés, et l'on en déduit les signatures expérimentales possibles, que l'on étudie par la suite.

Or l'expérimentateur fait exactement l'inverse : nous obtenons dans une expérience des signatures, donc un ensemble de points dans l'espace des signatures, et nous souhaitons remonter au modèle

sous-jacent qui explique ces signatures expérimentales. C'est ce que l'on appelle un scan inverse : de l'espace des signatures à l'espace des paramètres. La question immédiate qui se pose est : peut-on remonter univoquement d'un ensemble de signatures expérimentales (un point dans l'espace des signatures) à un modèle sous-jacent (un point dans l'espace des paramètres) ?

On peut avoir le cas simple où c'est possible :

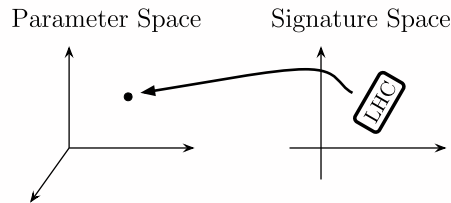


FIG. 20 – Cas idéal du scan inverse

et le cas moins simple, où justement ce scan n'est pas univoque :

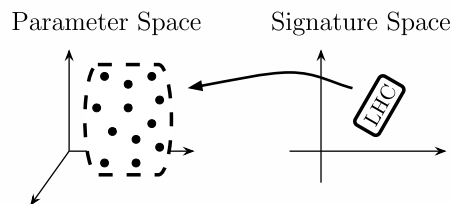


FIG. 21 – Cas plus réaliste du scan inverse, où des dégénérescences apparaissent

Cette question est essentielle : nous souhaitons identifier précisément quel type de nouvelle physique est potentiellement découverte, et si une dégénérescence apparaît, il est crucial de la quantifier. C'est dans ce cadre que se pose le problème inverse du LHC : a-t'on des modèles dégénérés au LHC ?

## 4.2 L'étude du groupe de Harvard

### 4.2.1 Cadre de l'étude

L'étude du problème inverse du LHC a été réalisée par Arkani-Hamed, Kane, Thaler et Wang, que l'on va appeler dans la suite "groupe de Harvard". La référence de l'étude est [8].

La méthodologie est la suivante : sélectionner 43 026 points au hasard dans l'espace des paramètres du MSSM, puis générer pour chaque point des événements supersymétriques. À partir de maintenant un modèle supersymétrique est un point dans l'espace des paramètres, c'est-à-dire un ensemble de valeurs déterminées pour les divers paramètres du MSSM.

Le groupe de Harvard a ensuite comparé les signatures obtenues pour chaque modèle, ces signatures étant par exemple la distribution des jets, des photons émis, l'énergie manquante, etc... Le bruit de fond dû aux événements du Modèle Standard, notamment  $pp \rightarrow ZZ, WW$ , etc..., n'a pas été pris en compte, et la luminosité employée pour le générateur d'événement était de  $10 \text{ fb}^{-1}$ .

Le but est donc de savoir si l'on peut distinguer des modèles, et non pas de les identifier séparément.

### 4.2.2 Paramètres étudiés

15 paramètres indépendants ont été pris en compte, selon une distribution aléatoire uniforme. Voici la gamme de valeurs utilisée :

- $2 \leq \tan \beta \leq 50$
- $100 \text{ GeV} \leq M_1, M_2, \mu, m_{\tilde{L}_{1,2}}, m_{\tilde{E}_{1,2}}, m_{\tilde{L}_{13}}, m_{\tilde{\tau}_R} \leq 1 \text{ TeV}$
- $600 \text{ GeV} \leq M_3, m_{\tilde{Q}_{1,2}}, m_{\tilde{U}_{1,2}}, m_{\tilde{D}_{1,2}}, m_{\tilde{Q}_3}, m_{\tilde{t}_2}, m_{\tilde{b}_2} \leq 1 \text{ TeV}$

où L,E,Q,U,D, définissent respectivement les sleptons gauches, les sleptons droits, les squarks gauches, les squarks droits de type up ( $\tilde{u}, \tilde{c}, \tilde{t}$ ), les squarks droits de type down ( $\tilde{d}, \tilde{s}, \tilde{b}$ ). La masse du boson de Higgs  $A^0$  est mise à 800 GeV pour toute l'étude, les termes triliénaires liés à la troisième génération de squarks sont mis à 850 GeV, et les termes trilinéaires des smuons et sélectrons sont mis à 0 GeV.

Ce choix permet d'avoir une section efficace totale large au LHC pour la supersymétrie, tout en ayant des gluinos et des squarks de masse assez élevée pour éviter que la section efficace ne soit trop importante, afin que le calcul informatique résiste.

### 4.2.3 Résultats : il existe un problème inverse du LHC !

L'étude du groupe de Harvard, utilisant la méthodologie décrite ci-dessus sur l'ensemble des 43 026 modèles utilisés, a dénombré 286 paires de modèles indiscernables au LHC, qui mettent en jeu 385 modèles différents - des modèles pouvant se trouver dans plus d'une paire. Cela semble une bonne chose finalement, puisque cela fait moins d'1 %.

Cependant, le résultat qui nous intéresse le plus est de savoir le nombre *total* de modèles indiscernables d'un modèle générique. Or ici l'étude n'a scanné qu'une partie de l'espace des paramètres : pour estimer l'ordre de grandeur de la dégénérescence, on utilise le fait que l'on peut estimer statistiquement cette dégénérescence selon le même principe que la probabilité des paires d'anniversaires au sein d'un groupe de personnes.

De cette manière, le groupe de Harvard a annoncé qu'il y a environ  $3,3 \cdot 10^6$  signatures différentes du MSSM auprès du LHC. Cela nous donne une dégénérescence de l'ordre de 10 à 100 modèles pour un modèle générique donné, avec des critères de similitude pourtant assez larges - par exemple si l'on considère des modèles où la seule différence est sur la masse de leurs gauginos que l'on autorise à varier de 10 % autour d'une valeur donnée, on a une dégénérescence moyenne de 12,6.

On a donc bien affaire à un problème inverse avec le LHC ! Cette étude a pourtant été réalisée sans bruit de fond dû au Modèle Standard, on s'attend donc que pour une étude plus fine avec bruit de fond, le résultat soit encore pire...

## 4.3 Étude sur l'ILC et création d'un fichier de contraintes expérimentales

### 4.3.1 L'ILC et le problème inverse du LHC

La question que notre équipe se pose est la suivante : l'ILC peut-il résoudre le problème inverse du LHC vu ci-dessus, du fait de sa plus grande précision dans les mesures ? En effet, l'ILC étant un collisionneur leptonique, on contrôle parfaitement l'état initial au contraire d'un collisionneur hadronique, où les collisions mettent en jeu des sous-constituants des protons, ce qui fait que l'état initial n'est pas parfaitement déterminé et qui explique pourquoi nous ne travaillons que dans le mode transverse.

Le principal objectif de notre groupe est donc de reprendre l'étude de Harvard, en y ajoutant de plus le bruit de fond du Modèle Standard, et voir si le problème inverse du LHC se résout. L'étude

prend l'ILC avec une énergie du centre de masse de 500 GeV. La luminosité est de  $500 \text{ fb}^{-1}$ .

Ce premier objectif en a ensuite entraîné un second, qui est celui sur lequel cette partie du stage se concentre spécifiquement. En effet, on peut se poser la question suivante : dans quelle mesure les modèles indiscernables de l'étude du groupe de Harvard sont-ils réalistes d'un point de vue physique ? Une grande partie d'entre eux a des chances d'être en définitive déjà exclue par l'expérience.

Pour cela, notre but est de créer un ensemble de contraintes expérimentales le plus large possible, associant contraintes de physique des particules et contraintes astrophysiques et cosmologiques. Il faut donc avoir un programme qui génère au hasard des points dans l'espace des paramètres, calcule le spectre de masse de ces points puis un ensemble de données qui seront soumis aux contraintes expérimentales pour savoir si ou non le modèle est réaliste. Puis nous pourrons ensuite en tirer les informations phénoménologiques et savoir s'ils sont ou non discernables à l'ILC.

Nous utilisons un ensemble de 15 paramètres, qui reprend dans les grandes lignes l'ensemble du groupe de Harvard. Nous prenons des masses de squarks égales pour tous les quarks de toute chiralité, excepté le stop ; cela étant une conséquence des études réalisées sur les mélanges  $K - \bar{K}$  et  $D - \bar{D}$ . Nous distinguons le stop gauche et le stop droit, ainsi que les sleptons gauches et droits.  $M_3$  est fixé à 1 TeV.

### 4.3.2 Contraintes astrophysiques et cosmologiques

Les contraintes astrophysiques et cosmologiques prises en compte concernent la matière noire. Elles sont de deux types : contraintes sur la densité relique  $\Omega_{CDM}h^2$  qui concerne la recherche indirecte de matière noire, et contraintes sur la détection directe de WIMP à l'aide de l'expérience XENON [6] qui est celle qui a fournit les résultats les plus récents (avril 2007).

Nous avons ainsi récupéré les données de XENON10, puis nous les avons ajustées pour avoir une équation de la courbe décrivant l'évolution de la section efficace de diffusion WIMP-nucléons en fonction de la masse des WIMP ici prise comme masse du neutralino le plus léger, équation que nous allons utiliser dans le programme de contraintes. Voici l'ajustement et l'erreur commise dessus :

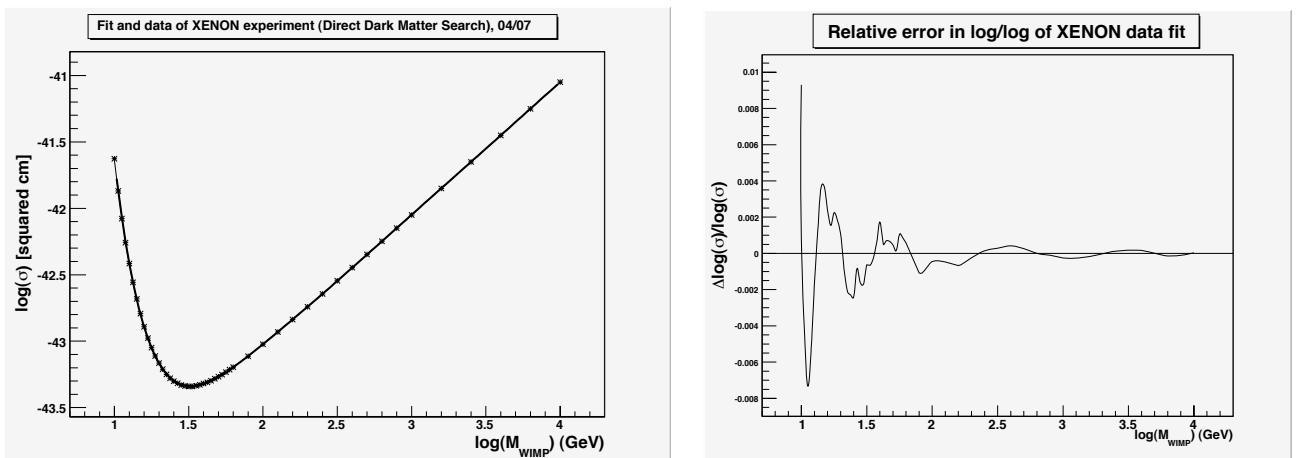


FIG. 22 – Évolution de  $\sigma(\tilde{\chi}_1^0 + p, n \rightarrow \tilde{\chi}_1^0 + p, n)$  en fonction de la masse de  $\tilde{\chi}_1^0$ , et erreur sur l'ajustement à droite. Ajustement  $\log_{10}(\sigma) = 35,787621 - 226,638876 \log(M_{\tilde{\chi}_1^0}) + 278,145916 \log^2(M_{\tilde{\chi}_1^0}) - 192,145996 \log^3(M_{\tilde{\chi}_1^0}) + 81,962485 \log^4(M_{\tilde{\chi}_1^0}) - 22,088651 \log^5(M_{\tilde{\chi}_1^0}) + 3,671239 \log^6(M_{\tilde{\chi}_1^0}) - 0,343998 \log^7(M_{\tilde{\chi}_1^0}) + 0,013913 \log^8(M_{\tilde{\chi}_1^0})$

L'erreur est donc négligeable dans la gamme qui nous intéresse à savoir  $m_{\tilde{\chi}_1^0} > 45$  GeV qui est compatible avec la limite sur la détection directe du neutralino. Ainsi, lorsque l'on compare les sections efficaces de nos modèles avec les données, le modèle est valide si  $\sigma_{\text{modèle}} < \sigma_{\text{données}}$ ,  $\sigma_{\text{données}}$  étant calculé à l'aide de l'ajustement de la courbe pour la masse du neutralino du modèle étudié.

Pour ce qui est de la limite prise sur la densité de matière noire, nous utilisons les données de WMAP3 [1], ce qui donne  $\Omega_{CDM} \leq 0.1211$  à 95 % CL.

### 4.3.3 Contraintes liées aux mesures en accélérateur

Les contraintes liées aux mesures expérimentales obtenues auprès des accélérateurs sont de deux types : soit elles sont dépendantes du modèle utilisé pour décrire la brisure de supersymétrie (modèle mSUGRA le plus souvent), soit elles s'inscrivent dans un cadre plus général au sein du MSSM. Il va de soit que seul le second type de résultats nous intéresse.

Une étude détaillée des résultats fournis par le Tevatron montre que bon nombre de contraintes sur les masses des superparticules autres que les neutralinos et charginos sont dépendantes du modèle choisi, notamment en ce qui concerne les gluinos. C'est pourquoi leur étude a été reportée à plus tard.

Nous avons donc étudié les contraintes qui s'inscrivent dans le cadre du MSSM, et nous les avons dans un premier temps codées sous DarkSUSY [17]. Voici l'ensemble des contraintes utilisées dans notre étude :

- contrainte sur le paramètre  $\Delta\rho$  : ce paramètre quantifie la brisure électrofaible  $SU(2)$ . Dans le Modèle Standard on a  $\rho = 1$ , des effets de nouvelle physique, notamment dans le cadre de la supersymétrie la présence d'un second doublet de Higgs, peuvent faire varier ce paramètre. D'après les données du PDG [10], on a  $-0,0007 \leq \Delta\rho \leq 0,0026$  à 95 % CL.
- contrainte sur  $b \rightarrow s\gamma$  : ce changement de saveur est très sensible à la nouvelle physique via les corrections radiatives. On utilise la valeur du taux de branchement de cette désintégration, qui provient des expériences BaBar et Belle. Un calcul nous donne  $BR(b \rightarrow s\gamma) = (0,47 \pm 0,78) \cdot 10^{-4}$  à 95 % CL.
- contrainte sur  $(g - 2)_\mu$  : le facteur de Landé du muon est une contrainte très sévère sur toute nouvelle physique, et actuellement il y a une différence entre le Modèle Standard et l'expérience qui ne se recoupe que dans les incertitudes. Nous prenons les données récentes fournies dans [13] ; nous choisissons d'étudier la différence entre Modèle Standard et expérience, et nous prenons comme incertitude supérieure celle donnée par l'expérience, comme incertitude inférieure celle donnée par le Modèle Standard. On obtient alors la contrainte suivante sur  $a_\mu = \frac{(g - 2)_\mu}{2}$  :  $-12,6 \cdot 10^{-10} \leq \Delta a_\mu \leq 40,1 \cdot 10^{-10}$ .
- contrainte sur la largeur invisible du  $Z^0$  : ce boson peut se désintégrer de plusieurs manières, dont l'une d'elles est invisible car donnant des particules non chargées. Entre autre au sein du Modèle Standard il y a la désintégration en neutrinos via interaction faible. La supersymétrie offre de nouveaux canaux de désintégration grâce aux neutralinos, cette largeur contraint donc la supersymétrie. Le PDG (10) nous donne  $\Gamma_{SUSY} \leq 2,0$  MeV à 95 % CL.
- bosons de Higgs : nous reprenons les limites déjà données par le PDG [10], et compte tenu de la sensibilité du calcul du spectre, nous avons choisi de réduire la limite sur le boson de Higgs léger à  $m_{h^0} \geq 112$  GeV et non 114,4 GeV. Ce n'est qu'après un premier filtrage avec les autres contraintes que nous renforcerons celle liée au boson  $h^0$ .
- charginos et neutralinos : nous reprenons ici les contraintes fournies par le LEP, résumées par le PDG [10].

Nous utilisons aussi la nouvelle valeur de la masse du quark top, publiée par le Particle Data

Group ce printemps [10] :  $M_t = (170,9 \pm 1,8)$  GeV.

Notre travail a donc été de choisir et de coder ces contraintes dans DarkSUSY.

#### 4.3.4 Problèmes rencontrés et modèles exclus

Les premiers tests effectués notamment sur les modèles génériques SPS1a puis SPS1a' [3] ont montré des failles dans les calculs effectués par DarkSUSY, par rapport à des résultats plus récents. En effet, les calculs notamment sur  $\Omega_{CDM}h^2$  n'atteignent pas la précision demandée et sont assez loin des résultats déjà publiés : sur SPS1a' par exemple le calcul fournit  $\Omega_{CDM}h^2 = 0,181 > 0,1211$ , en contradiction avec (1). Ce problème a été confirmé après discussion avec d'autres équipes, et il se trouve que les auteurs du programme travaillent actuellement à une modification du programme pour rectifier cette erreur.

De plus, DarkSUSY est un programme assez vieux (2004) et toutes ses routines ne sont pas optimisées pour le calcul du spectre en masse. Enfin, il apparaît après une étude de code que DarkSUSY n'utilise pas les masses physiques en entrée, mais les masses de renormalisation, ce qui n'est pas ce que nous souhaitons utiliser. Malgré tous ces défauts, il semblerait quand même que le modèle SPS1a' soit exclu grâce à la nouvelle masse du top. Mais une étude plus approfondie et débarassée de ces soucis est à conduire avant de pouvoir l'affirmer avec certitude.

#### 4.3.5 La solution avec une interface Isajet-DarkSUSY couplée à Micromegas ?

Afin de s'affranchir des nombreux problèmes rencontrés avec DarkSUSY, nous avons opté pour un autre logiciel dont la dernière mise à jour date de février 2007 : Isajet. Ce logiciel est très souvent utilisé pour les études en collisionneur, notamment par les équipes qui travaillent sur LHC. Nous avons décidé de l'utiliser pour calculer le spectre en masse de nos modèles, puis de construire une interface entre ce logiciel et notre code tournant sous DarkSUSY, afin d'y calculer notamment  $BR(b \rightarrow s\gamma)$  ou les contraintes sur la recherche directe de matière noire.

Nous avons ainsi créé un programme qui calcule le spectre en masse de nos modèles à partir d'un ensemble de paramètres pris à l'échelle électrofaible, puis nous avons créé une interface entre IsaSUSY (la partie d'Isajet qui calcule le spectre) et DarkSUSY.

Pour le calcul de  $\Omega_{CDM}h^2$  nous nous sommes tournés vers un programme récent et qui semble de confiance, appelé Micromegas. Nous avons réalisé une étude comparative entre le spectre calculé au sein de Micromegas et celui calculé avec IsaSUSY : la différence n'excède pas 1% dans le pire des cas, ce qui semble être un bon résultat mais ne nous affranchit pas en définitive de la création d'une interface IsaSUSY/Micromegas, ce qui a été réalisé dans les dernières semaines de ce stage.

La procédure adoptée est donc la suivante :

1. Utiliser IsaSUSY pour calculer le spectre de masse au sein du MSSM, ainsi que les paramètres de mélanges.
2. Utiliser l'interface IsaSUSY/DarkSUSY pour calculer les limites sur la recherche directe de matière noire
3. Utiliser Micromegas avec le spectre d'IsaSUSY via l'interface créée, pour calculer  $a_\mu$ ,  $BR(b \rightarrow s\gamma)$  ainsi que  $\Delta\rho$

À chaque étape est utilisé le fichier de contraintes que nous avons élaboré.

## Conclusion

Ce stage s'est donc achevé le 28 août 2007, et m'a énormément apporté. J'ai pu tout d'abord découvrir la Californie et le mode de fonctionnement d'un laboratoire américain, au sein d'un environnement de travail assez exceptionnel. Ce fut aussi ma première expérience de recherche en physique théorique, expérience qui fut très fructueuse puisque elle m'a permis d'apprendre à manier les outils informatiques utilisés dans une étude théorique et m'a permis d'aborder la supersymétrie qui est un des candidats les plus prometteurs pour la nouvelle physique des hautes énergies de demain.

J'ai aussi eu la chance de participer à la genèse d'un tout nouveau projet, auquel j'ai pu pleinement m'investir. Ce projet a encore de nombreuses heures de travail devant lui, et je garderai bien sûr contact avec mon équipe pour être tenu au courant et pourquoi pas continuer à contribuer. Nous attendons avec impatience les résultats du LHC! Une publication est prévue à terme, mais elle n'interviendra certainement pas avant au moins 1 an, vu l'ampleur de la tâche qu'il nous reste à accomplir. Il faut notamment réfléchir à comment parcourir l'espace des paramètres, se préoccuper des limites sur la détection directe des gluinos, etc...

Enfin ce stage a aussi été l'occasion de découvrir la Californie, comme en témoigne cette photo de Yosemite :





## Références

- [1] Edward Baltz, Marco Battaglia, Michael Peskin, and Tommer Wizansky, *Determination of Dark Matter Properties at High Energy Colliders*, SLAC-PUB-11687 (2006).
- [2] C-H Chen, M. Drees, and JF Gunion, "Searching for invisible and almost invisible particles at  $e+e-$  colliders", UCD-95-918 (1999), [hep-ph/9512230v3](#).
- [3] Aguilar-Saavedr et al., *Supersymmetry Parameters Analysis : SPA Convention and Project*, SLAC-PUB-11579 (2005).
- [4] Allanach et al., *The Snowmass Points and Slopes : Benchmarks for SUSY searches*, Eur. Phys. J.C. **25** (2002), 113, SLAC-PUB 751.
- [5] Angle et al., *First results from the XENON10 Dark Matter Experiment at the Gran Sasso National Laboratory*, 2007, [arXiv :0706.0039v1](#).
- [6] H. Baer et al., *Searching for Supersymmetry at  $e^+e^-$  supercolliders*, Int. J. Mod. Phys. A **4** (1989), 4111.
- [7] Manuel Drees et al., *Theory and Phenomenology of Sparticles*, World Scientific, 2004.
- [8] N. Arkani-Hamed et al., *Supersymmetry and the LHC inverse problem*, 2005, [hep-th/0512190](#).
- [9] S. Mrenna et al., *Pythia 6.4 : Physics and Manual*, 2006, [hep-ph/0603175](#).
- [10] W.-M. Yao et al (Particle Data Group), *Review of Particle Physics 2006 and 2007 partial update*, J. Phys. G **33** (2006), 1.
- [11] Jonathan L. Feng and Mihoko M. Nojiri, *Supersymmetry and the Linear Collider*, UCI-TR-02-37 (2002).
- [12] G. J. Gounaris, J. Layssac, P. I. Porfyriadis, and F. M. Renard, *Neutralino pair production at CERN LHC*, Phys. Rev. D **70** (2004), 033011.
- [13] M. Davier , *The Hadronic Contribution to  $(g - 2)_\mu$* , [arXiv :hep-ph/0701163v2](#)
- [14] H. Haber and G. Kane, *The search for supersymmetry : probing the physics beyond the Standard Model*, Physics Letters **117** (1985), 75–263.
- [15] Stefen P. Martin, *A supersymmetry primer*, 1997, [hep-ph/9709356](#).
- [16] Michael Peskin and Daniel Schroeder, *An Introduction to Quantum Field Theory*, Westview, 1995.
- [17] P. Gondolo et al., *Darksusy : Computing supersymmetric dark matter properties numerically*, JCAP **008** (2004), [astro-ph/0406204](#).
- [18] Julius Wess and Jonathan Begger, *Supersymmetry and Supergravity*, Princeton series in Physics, 1992.

## A Programme sur les bornes expérimentales

Listing 1 – dsacbd7.f - parties modifiées du programme principal

```

1  c -----limit on the rho parameter
  c ... je addition 2000-04-24
  c ... uses delta rho as calculated by feynhiggs
  c     if (delrho.gt.1.3d-3) then
  c         excl=ibset(excl,8)
6  c     endif

  c ... New value, pdg2007, at 95 % CL :
  write(*,*) 'Deltarho_=_', delrho
  if (delrho.gt.2.6d-3.or.delrho.lt.-7.0e-4) then
11     excl=ibset(excl,8)
  end if

  c -----limit on (g-2) muon
16 c ... Please refer to hep-ph/0701163 (Davier)
  c ... We take two sigma above experiment and two sigma under SM prediction
  c ... of (g-2)/2
  gggmuon=dsgm2muon()
  write(*,*) 'g-2_muon_=_', gggmuon
21  if (gggmuon.lt.-12.6e-10.or.gggmuon.gt.40.1e-10) then
  excl=ibset(excl,9)
  end if

  c -----limit on dark matter density
26 c ... WMAP3 data, astro-ph/0603449
  c ... We take 95 % CL
  oh2=dsrdomega(1,1,xf,ierr,iwar,nfc)
  write(*,*) 'Relic_density_=_', oh2
  if (oh2.gt.0.1211) then
31     excl=ibset(excl,10)
  end if

  c -----limit on dark matter direct search
  c ... XENON10 experiment, arXiv:0706.0039v1, astro-ph
36  call dsddneunuc(sigmap,sigman,otherp,othern)
  sigmaxenon = (sigmap*54.0+sigman*77.0)/131.0
  xenondat = log(mass(kn(1)))/log(10.0)
  call darkmatterxenon(xenondat,testxenon)
  sigmaxenon=log(sigmaxenon)/log(10.0)
41  write(*,*) 'cross_section_calculated_=_', sigmaxenon
  write(*,*) 'data_=_', testxenon
  if (sigmaxenon.gt.testxenon) then
  excl=ibset(excl,11)
  end if

```

```

c      reason for exclusion
1000 continue
      if (prtlevel.ge.1) call dswexcl(6,excl)
      return
51
end

subroutine darkmatterxenon(x, res)
real*8 x, res
56  res= 35.787621-226.638876*x + 278.145916*x**2 - 192.145996*x**3+
&      81.962485*x**4 -22.088651*x**5 + 3.671239*x**6-
&      0.343998*x**7+ 0.01391*x**8
      return
end

```

Listing 2 – dsacbnd.f - programme de gestion des contraintes expérimentales

```

subroutine dsacbnd(excl)
implicit none
include 'dsacom.h'
integer excl
5  if (aclabel.eq.'pdg2007'.or.aclabel.eq.'default') then
      call dsacbnd7(excl)
elseif (aclabel.eq.'pdg2002c') then
      call dsacbnd6(excl)
elseif (aclabel.eq.'pdg2002b') then
10  call dsacbnd5(excl)
else if (aclabel.eq.'pdg2002') then
      call dsacbnd4(excl)
else if (aclabel.eq.'pdg2000') then
      call dsacbnd3(excl)
15  else if (aclabel.eq.'mar2000') then
      call dsacbnd2(excl)
else if (aclabel.eq.'pdg1999') then
      call dsacbnd1(excl)
else
20  write(*,*) 'Error in dsacbnd: invalid option:',aclabel
      stop
endif
return
end

```

Listing 3 – dswexcl.f - programme de gestion des erreurs

```

1  subroutine dswexcl(unit, excl)
c -----
c  write reasons for exclusion to specified unit.
c  input:
c    unit - logical unit to write on (integer)
6  c    excl - code of the reason for exclusion (integer); 0 if allowed
c    author: paolo gondolo (gondolo@lpthe.jussieu.fr) 1994
c -----

```

```

implicit none
include 'dssusy.h'
11 integer unit, excl
character*12 dsidtag
if (unit.le.0) return
if (excl.eq.0) return
write (unit,1001) dsidtag(), '_excluded_by:_'
16 if (btest(excl,0)) write (unit,1000) 'chargino_mass;_'
if (btest(excl,1)) write (unit,1000) 'gluino_mass;_'
if (btest(excl,2)) write (unit,1000) 'squark_mass;_'
if (btest(excl,3)) write (unit,1000) 'slepton_mass;_'
if (btest(excl,4)) write (unit,1000) 'gamma_z(inv);_'
21 if (btest(excl,5)) write (unit,1000) 'h2_mass;_'
if (btest(excl,6)) write (unit,1000) 'neutralino_mass;_'
if (btest(excl,7)) write (unit,1000) 'b->s+gamma;_'
if (btest(excl,8)) write (unit,1000) 'delta_rho;_'
if (btest(excl,9)) write (unit,1000) '(g-2)/2_muon;_'
26 if (btest(excl,10)) write (unit,1000) 'relic_density;_'
if (btest(excl,11)) write (unit,1000) 'direct_search;_'
write (unit,*)
1000 format (1x,a,$)
1001 format (1x,'(',a,')',1x,a,$)
31 end

```

## B Interface Isajet/DarkSUSY et Isajet/Micromegas

Listing 4 – maincode-isa.f - génération du spectre par IsaSUSY

```

PROGRAM MAINCODE

IMPLICIT NONE

4
C ISASUSY common blocks
COMMON /GOOD/ NNOGOOD,IALLOW
INTEGER NNOGOOD,IALLOW
INTEGER IMODIN,IMODEL
9 REAL*4 xxmg,xxmu,xxmha,xxtanb,xxmq1,xxmdr,xxmur,
& amtmp,
& xxml1,xxmer,xxmq2,xxmsr,xxmer,xxml2,xxmmr,xxmq3,
& xxmbr,xxmtr,xxml3,xxmlr,xxat,xxab,xxal,xxm1,xxm2,xxmt

14 include 'sspar.h'
include 'ssmode.h'
include 'sssm.h'
include 'id.h'

19 REAL PI
CHARACTER*30 ISAPW2
SAVE ISAPW2

```

C

```

24 C ISAJET
CHARACTER*40 VERSN, VISAJE
include 'sslun.h'

29 C
CHARACTER*6 PINF

C

34 CHARACTER*80 FNAME, FNLHA, FNWIG
LOGICAL GOLHA, GOWIG
INTEGER ILHA, IWIG
REAL XLAMGM, XMESGM, XN5GM, XCMGV

39 LOUT=1

call load_par (20, xxmg, xxmu, xxmha, xxtanb, xxmq1, xxmdr, xxmur,
& amtmp, xxml1, xxmer, xxmq2, xxmsr, xxmcr, xxml2, xxmmr, xxmq3,
& xxmbr, xxmtr, xxml3, xxmlr, xxat, xxab, xxal, xxm1, xxm2, xxmt)

44 IMODEL=1
IALLOW=1

OPEN(UNIT=12, FILE="isa_spectrum.txt")

49 OPEN(UNIT=20, FILE="isa2micro.txt")

PI=4.*ATAN(1.)
QSUSY=SQRT(xxmq3*xxmtr)
54 ALQCD4=0.177
AMBT=AMASS(5)
AMTP=xxmt
ASMB=SUALFS(AMBT**2, .36, AMTP, 3)
MBMB=AMBT*(1. - 4*ASMB/3. / PI)
59 MBQ=SNGL(SSMQCD(DBLE(MBMB), DBLE(QSUSY)))
ASMT=SUALFS(AMTP**2, .36, AMTP, 3)
MIMT=AMTP/(1. + 4*ASMT/3. / PI + (16.11 - 1.04*(5. - 6.63/AMTP))*
$ (ASMT/PI)**2)
MITQ=SNGL(SSMQCD(DBLE(MIMT), DBLE(QSUSY)))
64 MLQ=1.7463
C For MSSM solution TANBQ=TANB; for SUGRA, TANBQ=/ TANB
ALFAEM=1./128.
SN2THW=.232
AMW=80.423
69 GG=SQRT(4*PI*ALFAEM/SN2THW)
VUQ=SQRT(2*AMW**2/GG**2/(1.+1./xxtanb**2))
VDQ=VUQ/xxtanb

CALL SSMSSM(xxmg, xxmu, xxmha, xxtanb, xxmq1,

```

```

74      &      xxmdr , xxmur , xxm11 , xxmer , xxmq2 ,
      &      xxmsr , xxmcr , xxm12 , xxmmr , xxmq3 ,
      &      xxmbr , xxmtr , xxm13 , xxmlr , xxat ,
      &      xxab , xxal , xxm1 , xxm2 ,
      &      xxmt , IALLOW , IMODEL)

79      WRITE(12 , *) 'M1_mass_:'
      WRITE(12 , *) xxm1
      WRITE(12 , *) 'M2_mass_:'
      WRITE(12 , *) xxm2
84      WRITE(12 , *) 'gluino_MASS_:'
      WRITE(12 , *) AMGLSS

      WRITE(12 , *) 'higgsino_parameter_mu_:'
      WRITE(12 , *) TWOM1
89      WRITE(12 , *) 'ratio_v2/v1_of_VEV_:'
      WRITE(12 , *) 1.0/RV2V1

      WRITE(12 , *) 'up_left-right_squark_mass_:'
      WRITE(12 , *) AMULSS , AMURSS
94      WRITE(12 , *) 'down_left-right_squark_mass_:'
      WRITE(12 , *) AMDLSS , AMDRSS
      WRITE(12 , *) 'left , right_sbottom_masses_:'
      WRITE(12 , *) AMBLSS , AMBRSS
      WRITE(12 , *) 'light , heavy_sbottom_masses_:'
99      WRITE(12 , *) AMB1SS , AMB2SS
      WRITE(12 , *) 'left , right_stop_masses_:'
      WRITE(12 , *) AMTLSS , AMTRSS
      WRITE(12 , *) 'light , heavy_stop_masses_:'
      WRITE(12 , *) AMT1SS , AMT2SS

104     WRITE(12 , *) 'left_selectron_mass_:'
      WRITE(12 , *) AMELSS
      WRITE(12 , *) 'right_selectron_mass_:'
      WRITE(12 , *) AMERSS
109     WRITE(12 , *) 'left , right_stau_masses_:'
      WRITE(12 , *) AMLLSS , AMLRSS
      WRITE(12 , *) 'light , heavy_stau_masses_:'
      WRITE(12 , *) AML1SS , AML2SS
      WRITE(12 , *) 'sneutrino_mass_generation_1_:'
114     WRITE(12 , *) AMN1SS
      WRITE(12 , *) 'sneutrino_mass_generation_2_:'
      WRITE(12 , *) AMN2SS
      WRITE(12 , *) 'sneutrino_mass_generation_3_:'
      WRITE(12 , *) AMN3SS

119     WRITE(12 , *) 'stop_mixing_angle_:'
      WRITE(12 , *) THETAT
      WRITE(12 , *) 'sbottom_mixing_angle_:'

```

```

124  WRITE(12,*) THETAB
      WRITE(12,*) 'stau_mixing_angle:'
      WRITE(12,*) THETAL
      WRITE(12,*) 'stop_trilinear_term:'
      WRITE(12,*) AAT
129  WRITE(12,*) 'sbottom_trilinear_term:'
      WRITE(12,*) AAB
      WRITE(12,*) 'stau_trilinear_term:'
      WRITE(12,*) AAL

134  WRITE(12,*) 'neutral_Higgs_h0,H0,A0_masses:'
      WRITE(12,*) AMHL,AMHH,AMHA
      WRITE(12,*) 'charged_Higgs_H+_mass:'
      WRITE(12,*) AMHC

139  WRITE(12,*) 'Higgs_mixing_angle:'
      WRITE(12,*) ALFAH

      WRITE(12,*) 'signed_mass_of_Z1:'
      WRITE(12,*) AMZ1SS
144  WRITE(12,*) 'signed_mass_of_Z2:'
      WRITE(12,*) AMZ2SS
      WRITE(12,*) 'signed_mass_of_Z3:'
      WRITE(12,*) AMZ3SS
      WRITE(12,*) 'signed_mass_of_Z4:'
149  WRITE(12,*) AMZ4SS

      WRITE(12,*) 'neutralino_mixing_angles,_4*4:'
      WRITE(12,*) ZMIXSS(1,1),ZMIXSS(1,2),ZMIXSS(1,3),ZMIXSS(1,4)
      WRITE(12,*) ZMIXSS(2,1),ZMIXSS(2,2),ZMIXSS(2,3),ZMIXSS(2,4)
154  WRITE(12,*) ZMIXSS(3,1),ZMIXSS(3,2),ZMIXSS(3,3),ZMIXSS(3,4)
      WRITE(12,*) ZMIXSS(4,1),ZMIXSS(4,2),ZMIXSS(4,3),ZMIXSS(4,4)
      WRITE(12,*) 'chargino_left,right_mixing_angles:'
      WRITE(12,*) GAMMAL,GAMMAR

159  WRITE(12,*) 'W_mass_needed_for_DarkSUSY:'
      WRITE(12,*) AMW

      WRITE(12,*) 'signed_mass_of_W1:'
      WRITE(12,*) AMW1SS
164  WRITE(12,*) 'signed_mass_of_W2:'
      WRITE(12,*) AMW2SS

      CLOSE(12)

169  WRITE(20,*) xxtanb
      WRITE(20,*) xxmg
      WRITE(20,*) xxmu
      WRITE(20,*) xxmha
      WRITE(20,*) VUQ/VDQ

```

```

174    WRITE(20,*) xxmq1
      WRITE(20,*) xxmdr
      WRITE(20,*) xxmur
      WRITE(20,*) xxml1
      WRITE(20,*) xxmer
179    WRITE(20,*) xxmq2
      WRITE(20,*) xxmsr
      WRITE(20,*) xxmcr
      WRITE(20,*) xxml2
      WRITE(20,*) xxmmr
184    WRITE(20,*) xxmq3
      WRITE(20,*) xxmbr
      WRITE(20,*) xxmtr
      WRITE(20,*) xxml3
      WRITE(20,*) xxmlr
189    WRITE(20,*) xxat
      WRITE(20,*) xxab
      WRITE(20,*) xxal
      WRITE(20,*) xxm1
      WRITE(20,*) xxm2
194    WRITE(20,*) VUQ,VDQ
      WRITE(20,*) AMGLSS
      WRITE(20,*) AMULSS
      WRITE(20,*) AMURSS
      WRITE(20,*) AMDLSS
199    WRITE(20,*) AMDRSS
      WRITE(20,*) AMSLSS
      WRITE(20,*) AMSRSS
      WRITE(20,*) AMCLSS
      WRITE(20,*) AMCRSS
204    WRITE(20,*) AMB1SS
      WRITE(20,*) AMB2SS
      WRITE(20,*) AMT1SS
      WRITE(20,*) AMT2SS
      WRITE(20,*) AMN1SS
209    WRITE(20,*) AMN2SS
      WRITE(20,*) AMN3SS
      WRITE(20,*) AMELSS
      WRITE(20,*) AMERSS
      WRITE(20,*) AMMLSS
214    WRITE(20,*) AMMRSS
      WRITE(20,*) AML1SS
      WRITE(20,*) AML2SS
      WRITE(20,*) -1.0*AMZ1SS
      WRITE(20,*) -1.0*AMZ2SS
219    WRITE(20,*) -1.0*AMZ3SS
      WRITE(20,*) -1.0*AMZ4SS
      WRITE(20,*) -1.0*AMW1SS
      WRITE(20,*) -1.0*AMW2SS
      WRITE(20,*) AMHL

```



```

224     WRITE(20,*) AMHH
        WRITE(20,*) AMHA
        WRITE(20,*) AMHC
        WRITE(20,*) ZMIXSS(1,1),ZMIXSS(1,2),ZMIXSS(1,3),ZMIXSS(1,4)
        WRITE(20,*) ZMIXSS(2,1),ZMIXSS(2,2),ZMIXSS(2,3),ZMIXSS(2,4)
229     WRITE(20,*) ZMIXSS(3,1),ZMIXSS(3,2),ZMIXSS(3,3),ZMIXSS(3,4)
        WRITE(20,*) ZMIXSS(4,1),ZMIXSS(4,2),ZMIXSS(4,3),ZMIXSS(4,4)
        WRITE(20,*) GAMMAL,GAMMAR
        WRITE(20,*) ALFAH
        WRITE(20,*) thetal
234     WRITE(20,*) thetat
        WRITE(20,*) thetab

        CLOSE(20)

239     END

```

Listing 5 – maincode-dark.f - calcul des contraintes par DarkSUSY avec le spectre d’IsaSUSY

```
PROGRAM MAINCODE
```

```
IMPLICIT NONE
```

4

---

*C.... DarkSUSY init*

```

9     real*8 oh2,xf,dsrdomega,oh2prime ! relic density
        character*80 message,scr
        logical first
        data first /.true./
        integer ii, jj, idum,nfc,iwar,ierr,unphys
        integer hwarning, excl,acceptable,kk, iend
14     double precision mz,gamma_inv,gzinv2,temp,
&      mi,mj,p2,ei,ej,gzij,mz2,smratio,susysratio
        real*8 m0min,m0max,m5min,m5max,a0min,a0max,tbmin,tbmax,
&      deltaratio
        real*8 dsrndlin,dsrndsgn
19     real*8 isavariabile,isavariabile2,isavariabile3 ! read from ISASUSY file
        real*8 neutralino_mix(4,4),tmpneutralino(4) ! neutralino mixing matrix
        real*8 chargino_l,chargino_r ! chargino mixing

        real*8 zgm,dsabsq,aux,masstau,masst,masst,mwtmp
24     integer i,j,g,valid,itmp
        complex*16 chaumxtmp(2,2),chavmxtmp(2,2)
        real*8 thetax,thetay,cotgl,cotgr,singl,singr,cosgl,cosgr
        real*8 cosbeta
        real*8 mscale,dsralph3,dsrcmq,mx
29     logical dsisnan
        character*80 garbage

        real*8 gggmuon,dsgm2muon ! g-2 muon

```

```

34      real*8 sigmap , sigman , otherp , othern , testxenon , sigmaxenon ! direct search
      real*8 xenondata ! direct search
C-----

c      include of dsusy directly as the compiler crashes

39 *      ! mode: fortran !
*#####*
*              i n c l u d e      f i l e              *
*#####*

44 *****
***      susy.h                      ***
***      this piece of code is needed as a separate file          ***
***      the rest of the code 'includes' susy.h                    ***
C-----C

49 c      author: paolo gondolo (gondolo@lpthe.jussieu.fr) 1994,1995
c      10-nov-95 complex vertex constants
c      modified by joakim edsjo (edsjo@teorfys.uu.se) 97-02-11
*      * general variables
*      * particle codes
54      integer knue , ke , knumu , kmu , knutau , ktau , ku , kd , kc , ks , kt , kb , kgamma ,
&      kw , kz , kgluon , kh1 , kh2 , kh3 , khc , ksnu , kse1 , kse2 , ksnumu , ksmu1 ,
&      ksmu2 , ksnutau , kstau1 , kstau2 , ksu1 , ksu2 , ksd1 , ksd2 , ksc1 , ksc2 ,
&      kss1 , kss2 , kst1 , kst2 , ksb1 , ksb2 , kn1 , kn2 , kn3 , kn4 , kcha1 , kcha2 ,
&      kgluin , kgold0 , kgoldc
59      parameter ( knue=1, ke=2, knumu=3, kmu=4, knutau=5, ktau=6)
      parameter ( ku=7, kd=8, kc=9, ks=10, kt=11, kb=12)
      parameter ( kgamma=13, kw=14, kz=15, kgluon=16)
      parameter ( kh1=17, kh2=18, kh3=19, khc=20)
      parameter ( ksnu=21, kse1=22, kse2=23, ksnumu=24, ksmu1=25,
64 &      ksmu2=26, ksnutau=27, kstau1=28, kstau2=29)
      parameter ( ksu1=30, ksu2=31, ksd1=32, ksd2=33, ksc1=34, ksc2=35,
&      kss1=36, kss2=37, kst1=38, kst2=39, ksb1=40, ksb2=41)
      parameter ( kn1=42, kn2=43, kn3=44, kn4=45, kcha1=46, kcha2=47,
&      kgluin=48)
69      parameter ( kgold0=49, kgoldc=50)
c      knu=(nue, numu, nutau)      kl=(e, mu, tau)      kqu=(u, c, t)      kqd=(d, s, b)
c      ksqu=(~u1, ~c1, ~t1, ~u2, ~c2, ~t2)      ksqd=(~d1, ~s1, ~b1, ~d2, ~s2, ~b2)
      integer kse(2), ksmu(2), kstau(2), ksu(2), ksd(2), ksc(2), kss(2),
&      kst(2), ksb(2), kn(4), kcha(2), knu(3), kl(3), kqu(3), kqd(3),
74 &      ksnu(6), ksl(6), ksqu(6), ksqd(6)
      character*8 pname(0:50)
      common /pacodes/ kse, ksmu, kstau, ksu, ksd, ksc, kss, kst, ksb, kn, kcha,
&      knu, kl, kqu, kqd, ksnu, ksl, ksqu, ksqd, pname
*      * program switches
79      integer higloop, neuloop, bsgqcd
      real*8 msquarks, msleptons
      common /switch/ msquarks, msleptons,
&      higloop, neuloop, bsgqcd

```

```

*      set sfermion masses by hand
84  real*8 massup1(3), massup2(3), thetamixu(3),
&      massdn1(3), massdn2(3), thetamixd(3),
&      masssn(3), masssl1(3), masssl2(3), thetamixsl(3)
common/sfermionmass/ massup1, massup2, thetamixu,
&      massdn1, massdn2, thetamixd,
89  &      masssn, masssl1, masssl2, thetamixsl
*      indirect rates variables
*      real*8 emuth, thmumax
*      common /indrates/emuth, thmumax
*      model parameters
94  real*8 ckms12, ckms23, ckms13, ckmdelta
real*8 tanbe, mu, m2, m1, m3, ma, mass2u(3), mass2q(3),
&      mass2d(3), mass2l(3), mass2e(3), asoftu(3), asoftd(3), asofte(3)
common /sckm/ ckms12, ckms23, ckms13, ckmdelta
common /spar/ tanbe, mu, m2, m1, m3, ma, mass2u, mass2q, mass2d,
99  &      mass2l, mass2e, asoftu, asoftd, asofte
*      useful global variables + roption
integer lsp, kln
real*8 pi, s2thw, sinthw, costhw, cosbe, sinbe,
&      cos2be, sin2be, zg, lgzg, delrho, alph3mz
104 character*5 roption
common /iuseful/ lsp, kln
common /ruseful/ pi, s2thw, sinthw, costhw,
&      cosbe, sinbe, cos2be, sin2be, zg, lgzg, delrho, alph3mz
common /cuseful/ roption
109 *      coupling constants
real*8 g2weak, gyweak, g3stro, alphem, alph3, yukawa(12),
&      lam1, lam2, lam3, lam4, lam5, lam6, lam7
common /couplingconstants/ g2weak, gyweak, g3stro, alphem,
&      alph3, yukawa, lam1, lam2, lam3, lam4, lam5, lam6, lam7
114 *      mass spectrum
real*8 mass(0:50)
real*8 mcmc, mbmb, mtmt
common /mspctm/ mass, mcmc, mbmb, mtmt
c      * masses at mb scale
119 c      real*8 mbb, mcb
c      common /massmb/mbb, mcb
*      decay widths
real*8 width(0:50), hdwidth(22,4)
common /widths/ width, hdwidth
124 *      partial decay widths
real*8 prtial(54)
common /partials/ prtial
*      vertices
complex*16 gl(50,50,50), gr(50,50,50)
129 common /vrtxs/ gl, gr
*      quantum numbers
real*8 ncolor(12), wiso3(12), echarg(12)
complex*16 ckm(3,3)

```

```

common /qnum/ ckm,ncolor ,wiso3 ,echarg
134 * mixings
    real*8 alpha ,mix_stop ,mix_sbot ,mix_stau
    complex*16 neunmx(4,4),chaumx(2,2),chavmx(2,2),
&    slulmx(3,3),sldlmx(6,3),sldrmx(6,3),
&    squlmx(6,3),squrmx(6,3),sqdlmx(6,3),sqdrmx(6,3)
139 common /mixing/ neunmx,chaumx,chavmx,
&    slulmx,sldlmx,sldrmx,
&    squlmx,squrmx,sqdlmx,sqdrmx,alpha,mix_stop,mix_sbot,mix_stau
    include 'dsio.h'
* msugra variables
144 real*8 m0var,mhfvar,a0var,tgbetavar,sgnmuvar
common/sugrainput/m0var,mhfvar,a0var,tgbetavar,sgnmuvar
* DarkSUSY version and root directory
character*50 dsversion
character*128 dsroot
149 common/dsver/dsversion,dsroot
c save common blocks
save /pacodes/,/switch/,/sckm/,/spar/,/iuseful/,/ruseful/,
&    /cuseful/,/couplingconstants/,/mspctm/,/widths/,/partials/,
&    /qnum/,/mixing/,/sugrainput/,/dsver/
154 ***
*****end of susy.h*****
    include 'dsidtag.h'
159 C This call initializes the DarkSUSY package. This call should
C be the first call in any program using DarkSUSY.
C
call dsinit
164
C


---


C The amount of output onto standard output can be controlled by the
C variable prtlevel. Setting prtlevel=0 suppresses all messages,
C except fatal errors that terminate the program. Setting prtlevel=1
169 C displays informational messages. Setting prtlevel to higher values
C displays more and more information. In this example, we set
C prtlevel=1 to have a message from dsreadpar. We reset it to 0
C later. Had we set prtlevel=2 here, we would also have messages for
C each variable which is (re)set in dsreadpar. Caution: if prtlevel
174 C is reset in dsreadpar, the new value applies to subsequent
C messages.
C
    prtlevel=2
C
179 C Selection of PDG2006 set of parameters
C
open (unit=10,file='pdg2006.dat',status='old')
call dsreadpar(10)

```

```

        close (10)
184
c      Open the file which contains ISASUSY parameters
        open (unit=10,file='isa_spectrum.txt',form = 'formatted',
&         status='old')
189
c...  set m1, m2 and m3
        read (10,*) garbage
        read (10,*) isavariabile
        m1=isavariabile
194
        read (10,*) garbage
        read (10,*) isavariabile
        m2=isavariabile
        read (10,*) garbage
        read (10,*) isavariabile
199
        m3=isavariabile

c...  set mu.
        read (10,*) garbage
        read (10,*) isavariabile
204
        mu = -1.0*isavariabile

c...  set tan(beta)
        read (10,*) garbage
        read (10,*) isavariabile
209
        tanbe=isavariabile

c..    Perhaps forgoteen

214
        ma = 417.66
        mass2u(3) = 400.43**2
        mass2q(3) = 484.88**2
        mass2d(3) = 515.77**2
        mass2l(3) = 180.51**2
219
        mass2e(3) = 110.46**2
        aoftu(3) = -588.69
        aoftd(3) = -971.34
        aofte(3) = -448.20

224
c... Now set up constants; we need this call already here
c... since we call dschasct below
        call dssuconst

229 c===== Sfermion Masses "1" = "L", "2" = "R" =====
c      squark sup left-right
        read(10,*) garbage
        read(10,*) isavariabile , isavariabile2

```

```

    mass(ksu1)=isavariabile
234    mass(ksu2)=isavariabile2
    c squark scharm left-right
    mass(ksc1)=isavariabile
    mass(ksc2)=isavariabile2
    c squark sdown left-right
239    read(10,*) garbage
    read(10,*) isavariabile , isavariabile2
    mass(ksd1)=isavariabile
    mass(ksd2)=isavariabile2
    c squark sstrange left-right
244    mass(kss1)=isavariabile
    mass(kss2)=isavariabile2
    c squark sbottom 1,2
    read(10,*) garbage
    read(10,*) isavariabile , isavariabile2
249    read(10,*) garbage
    read(10,*) isavariabile , isavariabile2
    mass(ksb1)=isavariabile
    mass(ksb2)=isavariabile2
    c squark stop 1,2
254    read(10,*) garbage
    read(10,*) isavariabile , isavariabile2
    read(10,*) garbage
    read(10,*) isavariabile , isavariabile2
    mass(kst1)=isavariabile
259    mass(kst2)=isavariabile2

    c selectron left , right
    read(10,*) garbage
    read(10,*) isavariabile
264    mass(kse1)=isavariabile
    read(10,*) garbage
    read(10,*) isavariabile
    mass(kse2)=isavariabile
    c smuon left , right
269    mass(ksmu1)=mass(kse1)
    mass(ksmu2)=mass(kse2)
    c stau 1,2
    read(10,*) garbage
    read(10,*) isavariabile , isavariabile2
274    read(10,*) garbage
    read(10,*) isavariabile , isavariabile2
    mass(kstau1)=isavariabile
    mass(kstau2)=isavariabile2

279 c sneutrinos
    read(10,*) garbage
    read(10,*) isavariabile
    mass(ksnue)=isavariabile

```

```

      read(10,*) garbage
284  read(10,*) isavariabile
      mass(ksnumu)=isavariabile
      read(10,*) garbage
      read(10,*) isavariabile
      mass(ksnutau)=isavariabile

```

289

*c Slepton and squark mass matrices at the weak scale*

```

      mass2q(3) = mass(kst(1))**2
      mass2u(3) = mass(kst(2))**2
294  mass2d(3) = mass(ksb(2))**2
      mass2l(3) = mass(kstau1)**2
      mass2e(3) = mass(kstau2)**2
      do i=1,2
          mass2q(i) = mass(ksu(1))**2
299  mass2u(i) = mass(ksu(2))**2
          mass2d(i) = mass(ksd(2))**2
          mass2l(i) = mass(kse(1))**2
          mass2e(i) = mass(kse(2))**2
      enddo

```

304

*c===== Sfermion Mixings =====*

```

      do g=1,3
          do i=1,3
309  slulmx(i,g) = cmplx(0.d0,0.d0)
          enddo
          do i=1,6
              sldlmx(i,g) = cmplx(0.d0,0.d0)
              sldrm(x,i,g) = cmplx(0.d0,0.d0)
314  squlmx(i,g) = cmplx(0.d0,0.d0)
              squrmx(i,g) = cmplx(0.d0,0.d0)
              sqdlmx(i,g) = cmplx(0.d0,0.d0)
              sqdrmx(i,g) = cmplx(0.d0,0.d0)
          enddo
319  slulmx(g,g) = cmplx(1.d0,0.d0)
          sldlmx(g,g) = cmplx(1.d0,0.d0)
          sldrm(g+3,g) = cmplx(1.d0,0.d0)
          squlmx(g,g) = cmplx(1.d0,0.d0)
          squrmx(g+3,g) = cmplx(1.d0,0.d0)
324  sqdlmx(g,g) = cmplx(1.d0,0.d0)
          sqdrmx(g+3,g) = cmplx(1.d0,0.d0)
      enddo

```

329

*c===== the mixing angles =====*

```

      read (10,*) garbage
      read (10,*) isavariabile
      mix_stop=isavariabile
      read (10,*) garbage

```

```

    read (10,*) isavvariable
334 mix_sbot=isavvariable
    read (10,*) garbage
    read (10,*) isavvariable
    mix_stau=isavvariable

339 c===== Soft Terms =====
    do i=1,2
        asoftd(i)=0.0d0
        asoftu(i)=0.0d0
        asofte(i)=0.0d0
344 enddo
    read (10,*) garbage
    read (10,*) isavvariable
    asoftd(3)=isavvariable
    read (10,*) garbage
349 read (10,*) isavvariable
    asoftu(3)=isavvariable
    read (10,*) garbage
    read (10,*) isavvariable
    asofte(3)=isavvariable

354 c... The matrices we now have would be the same as the ones from
c... DarkSUSY, except that  $q_{-1}$  and  $q_{-2}$  are sometimes flipped, i.e.
c... the mixing matrices
c... are multiplied by  $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ 
359 c... compared to if DarkSUSY would calculate these

c===== stau mixing =====
    sldlmx(3,3) = cmplx(cos(mix_stau),0.d0)
364 sldlmx(6,3) = cmplx(-sin(mix_stau),0.d0)
    sldrmx(3,3) = cmplx(sin(mix_stau),0.d0)
    sldrmx(6,3) = cmplx(cos(mix_stau),0.d0)

c===== stop mixing =====
369 squlmx(3,3) = cmplx(cos(mix_stop),0.d0)
    squlmx(6,3) = cmplx(-sin(mix_stop),0.d0)
    squrmx(3,3) = cmplx(sin(mix_stop),0.d0)
    squrmx(6,3) = cmplx(cos(mix_stop),0.d0)

374 c===== sbottom mixing =====
    sqdlmx(3,3) = cmplx(cos(mix_sbot),0.d0)
    sqdlmx(6,3) = cmplx(-sin(mix_sbot),0.d0)
    sqdrmx(3,3) = cmplx(sin(mix_sbot),0.d0)
    sqdrmx(6,3) = cmplx(cos(mix_sbot),0.d0)

379 c===== Gluino Mass =====
    mass(kgluin)=m3

```



```

c===== Higgs Masses=====
384   read(10,*) garbage
      read(10,*) isavARIABLE , isavARIABLE2 , isavARIABLE3
      mass(kh2)=isavARIABLE
      mass(kh1)=isavARIABLE2
      mass(kh3)=isavARIABLE3
389   read(10,*) garbage
      read(10,*) isavARIABLE
      mass(khc)=isavARIABLE

c===== sign reversed =====
394   read(10,*) garbage
      read(10,*) isavARIABLE
      alpha=-1.0*isavARIABLE

c===== Neutralino and Chargino Masses
399 c===== ISASUGRA orders the neutralino and chargino mass eigenvalues
      read(10,*) garbage
      read(10,*) isavARIABLE
      tmpneutralino(1) = isavARIABLE
      mass(kn1)=abs(isavARIABLE)
404   read(10,*) garbage
      read(10,*) isavARIABLE
      tmpneutralino(2) = isavARIABLE
      mass(kn2)=abs(isavARIABLE)
      read(10,*) garbage
409   read(10,*) isavARIABLE
      tmpneutralino(3) = isavARIABLE
      mass(kn3)=abs(isavARIABLE)
      read(10,*) garbage
      read(10,*) isavARIABLE
414   tmpneutralino(4) = isavARIABLE
      mass(kn4)=abs(isavARIABLE)

c===== Neutralino Mixing =====

419   read(10,*) garbage
      read(10,*) neutralino_mix(1,1), neutralino_mix(1,2),
      & neutralino_mix(1,3), neutralino_mix(1,4)
      read(10,*) neutralino_mix(2,1), neutralino_mix(2,2),
      & neutralino_mix(2,3), neutralino_mix(2,4)
424   read(10,*) neutralino_mix(3,1), neutralino_mix(3,2),
      & neutralino_mix(3,3), neutralino_mix(3,4)
      read(10,*) neutralino_mix(4,1), neutralino_mix(4,2),
      & neutralino_mix(4,3), neutralino_mix(4,4)

429   do 100 i=1,4
      do 200 j=1,4
      if (tmpneutralino(i).le.0.0d0) then
          neunmx(i,j)=cmplx(neutralino_mix(5-j,i),0.0d0)

```

```

        else
434         neunmx(i , j)=cmplx(0.0d0 , neutralino_mix(5-j , i))
        endif
200     continue
        do 300 j=1,2
            neunmx(i , j)=-neunmx(i , j)
439     300     continue
100     continue

    zg = dsabsq(neunmx(1,1))+dsabsq(neunmx(1,2))
    zgm = dsabsq(neunmx(1,3))+dsabsq(neunmx(1,4))
444    lgzg = log10(zg)-log10(zgm)
    kln = 1
    lsp = kn(kln)

```

*===== Chargino Mixing =====*

449 *c... This approach extracts the chargino matrices from ISASUGRA, but  
c... with wrong signs of the eigenvalues sometimes  
c... We will use the ISASUGRA values, but with the signs determined  
c... from DarkSUSY*

```

454     read(10,*) garbage
     read(10,*) chargino_l , chargino_r

     cotgl=1.0d0/tan(chargino_l)
459     cotgr=1.0d0/tan(chargino_r)

     singl=sin(chargino_l)
     singr=sin(chargino_r)
     cosgl=cos(chargino_l)
464     cosgr=cos(chargino_r)

     chaumxtmp(1,2)=singl
     chavmxtmp(1,2)=singr
     chaumxtmp(1,1)=-cosgl
469     chavmxtmp(1,1)=-cosgr
     chaumxtmp(2,2)=cosgl
     chavmxtmp(2,2)=cosgr
     chaumxtmp(2,1)=singl
     chavmxtmp(2,1)=singr

```

474 *c... Use the m<sub>W</sub> value of ISASUGRA and recalculate the chargino  
c... matrices*

```

     mwtmp=mass(kw)
479     read(10,*) garbage
     read(10,*) isavariabile
c... Use mw=79.9522095 from isasugra to be consistent
     mass(kw)=isavariabile

```

```

      call dschasct
484      mass(kw)=mwtmp

c... Now set use the isasugra matrices, but with the DarkSUSY signs
      do i=1,2
        do j=1,2
489          chaumx(i,j)=abs(chaumxtmp(i,j))
          &          *sign(1.0d0,dreal(chaumx(i,j)))
          chavmx(i,j)=abs(chavmxtmp(i,j))
          &          *sign(1.0d0,dreal(chavmx(i,j)))
        enddo
494      enddo

      read(10,*) garbage
      read(10,*) isavariabile
      mass(kcha2)=abs(isavariabile)
499      mass(kcha1)=abs(isavariabile)
      read(10,*) garbage
      read(10,*) isavariabile

      close(10)

504

      mass(0)=1.d10

c----- global constants
509

      mx=mass(kn(1))

c----- Check that the mass spectrum is OK
514      unphys=0
      hwarning=0
c... Check that neutralino is LSP
      do i=1,2
        if (mass(kcha(i)).lt.mx) unphys=ibset(unphys,0)
519      enddo
      do i=1,3
        if (mass(ksnu(i)).lt.mx) unphys=ibset(unphys,1)
      enddo
      do i=1,6
524        if (mass(ksl(i)).lt.mx) unphys=ibset(unphys,2)
        if (mass(ksqu(i)).lt.mx) unphys=ibset(unphys,3)
        if (mass(ksqd(i)).lt.mx) unphys=ibset(unphys,4)
      enddo
      if (mass(kgluin).lt.mx) unphys=ibset(unphys,5)
529

c... Check that masses are not NaN
      if (dsisnan(mass(kh1))) unphys=ibset(unphys,6)
      if (dsisnan(mass(kh2))) unphys=ibset(unphys,6)

```

```

534   if ( dsisnan ( mass ( kh3 ) ) ) unphys=ibset ( unphys , 6 )
      if ( dsisnan ( mass ( khc ) ) ) unphys=ibset ( unphys , 6 )
      do i=1,4
          if ( dsisnan ( mass ( kn ( i ) ) ) ) unphys=ibset ( unphys , 6 )
      enddo
      do i=1,2
539         if ( dsisnan ( mass ( kcha ( i ) ) ) ) unphys=ibset ( unphys , 6 )
      enddo
      do i=1,3
          if ( dsisnan ( mass ( ksnu ( i ) ) ) ) unphys=ibset ( unphys , 6 )
      enddo
544      do i=1,6
          if ( dsisnan ( mass ( ksl ( i ) ) ) ) unphys=ibset ( unphys , 6 )
          if ( dsisnan ( mass ( ksqu ( i ) ) ) ) unphys=ibset ( unphys , 6 )
          if ( dsisnan ( mass ( ksqd ( i ) ) ) ) unphys=ibset ( unphys , 6 )
      enddo
549      if ( dsisnan ( mass ( kgluin ) ) ) unphys=ibset ( unphys , 6 )

```

*c* 

---

 *reset running things*

```

554   mass(0)=1.d10

      mscale=2.d0*mass( lsp )
      alph3=dsralph3( mscale )
      g3stro=sqrt( 4.d0*pi*alph3 )
      aux = g2weak/ dsqrt( 2.d0 ) / mass( kw )
559   yukawa( ktau ) = aux*dsrmq( mscale , ktau ) / cosbe
      yukawa( kqu(2) ) = aux*dsrmq( mscale , kc ) / sinbe
      yukawa( kqu(3) ) = aux*dsrmq( mscale , kt ) / sinbe
      yukawa( kqd(3) ) = aux*dsrmq( mscale , kb ) / cosbe

```

564 *c* 

---

 *some useful vertices*  
*c* *to be checked with isajet*  
**call** dsvertx

*c* 

---

 *and particle widths*  
569 *c...* *Add Higgs widths and QCD correction to them and*  
*c* *to vertices (here will check with isajet)*  
**call** dshigwid

574 *c...* *Widths changed 020613 by Joakim Edsjo (will check with isajet)*  
width( kcha(1) ) = 1.0d0  
width( kcha(2) ) = 1.0d0  
width( kn(1) ) = 1.0d0  
width( kn(2) ) = 1.0d0  
579 width( kn(3) ) = 1.0d0  
width( kn(4) ) = 1.0d0

**do** i=1,6

```

        width(ksl(i))=0.2d0
584      width(ksnu(i))=0.2d0
        width(ksqu(i))=0.2d0
        width(ksqd(i))=0.2d0
enddo

589
c... Prepare for rate calculations
if (unphys.eq.0.and.valid.le.0) then
    call dsprep
endif

594
c... Experimental bounds :

call dsacset('default')
call dsacbnd(excl)

599
write(*,*) '___hwarning__=',hwarning
write(*,*) '_____unphys__=',unphys
write(*,*) '_____excl__=',excl

604 c if (hwarning.eq.0.and.unphys.eq.0.and.excl.eq.0) then ! Model OK
if (hwarning.eq.0.and.unphys.eq.0) then
c Model OK without accelerator bounds
    acceptable=0
else
609     acceptable=-1
end if

write(*,*) '___acceptable__=',acceptable

614
if (acceptable.ne.0) then
    call dswunph(6,unphys)
    call dswexcl(6,excl)
end if
if (hwarning.ne.0) then
619     call dswhwarn(6,hwarning)
end if
write(*,*) acceptable
if (acceptable.eq.0) then

624 c ----- Test of the interface : -----

    open(unit=18,file='dark_output.txt')

write(18,*) '___Neutralino__mass__=',mass(kn(1))
629 write(18,*) '___Gaugino__fraction__=',
    & dsabsq(neunmx(1,1))+dsabsq(neunmx(1,2))
write(18,*) '___H1__mass__=',mass(kh1),width(kh1)
write(18,*) '___H2__mass__=',mass(kh2),width(kh2)

```

```

634  write(18,*) 'H3 mass = ', mass(kh3), width(kh3)
write(18,*) 'H mass = ', mass(khc), width(khc)
write(18,*) 'Neutralino_2 mass = ', mass(kn(2))
write(18,*) 'Neutralino_3 mass = ', mass(kn(3))
write(18,*) 'Neutralino_4 mass = ', mass(kn(4))
write(18,*) 'Chargino mass = ', mass(kcha(1))
639  write(18,*) 'Chargino_2 mass = ', mass(kcha(2))
write(18,*) 'Gluino mass = ', mass(kgluin)
write(18,*) '~e_L mass = ', mass(ksl(1))
write(18,*) '~mu_L mass = ', mass(ksl(2))
write(18,*) '~tau_L mass = ', mass(ksl(3))
644  write(18,*) '~e_R mass = ', mass(ksl(4))
write(18,*) '~mu_R mass = ', mass(ksl(5))
write(18,*) '~tau_R mass = ', mass(ksl(6))
write(18,*) '~q_L mass = ', mass(ksu(1))
write(18,*) '~q_R mass = ', mass(ksu(2))
649  write(18,*) '~t1 mass = ', mass(kst(1))
write(18,*) '~t2 mass = ', mass(kst(2))
write(18,*) '~b1 mass = ', mass(ksb(1))
write(18,*) '~b2 mass = ', mass(ksb(2))

654  c-----bound from gamma_z(invisible)
      mz=mass(kz)
      mz2=mz*mz
      gzinv2=0
659  gamma.inv = 3.0d0 * (g2weak/2/costhw)**2 * mz/(24.d0*pi)

      do i=1,3
        temp = 1.d0-4.d0*mass(ksnu(i))**2/mz2
        if (temp.gt.0.d0) then
664          temp = temp**(1.5d0) *
&            (g2weak/2/costhw)**2 * mz/(48.d0*pi)
          gzinv2 = gzinv2 + temp
        endif
      enddo
669  ! neutralinos
      c... pg fixed june 30, 2000
      do i=1,4
        mi = abs(mass(kn(i)))
        do j=i,4
674          mj = abs(mass(kn(j)))
          p2 = (mz2-(mi+mj)**2)*(mz2-(mi-mj)**2)/(4*mz2)
          ei = (mz2-mj**2+mi**2)/(2*mz)
          ej = (mz2-mi**2+mj**2)/(2*mz)
          if (p2.gt.0.d0.and.ei.gt.0.0d0.and.ej.gt.0.0d0) then
679            gzij = g2weak/(2.d0*costhw)*
&              (conjg(neunmx(i,3))*neunmx(j,3)-
&              conjg(neunmx(i,4))*neunmx(j,4))
            temp=sqrt(p2)/(2.d0*pi*mz2) * (

```

```

&          dsabsq(gzij) * (ei*ej+p2/3.0d0)-
684 &          dreal(gzij**2)*mi*mj)
          if (i.eq.j) temp=0.5d0*temp
          gzinv2 = gzinv2 + temp
        endif
      enddo
689 enddo

write(18,*) ' Invisible width of Z in SM = ', gamma_inv
write(18,*) ' Invisible width of Z in new physics = ', gzinv2
write(18,*) ' Deltarho = ', delrho
694 gggmuon=dsgm2muon()
write(18,*) ' g-2 muon = ', gggmuon
call dsbsgammafull(smratio,0)
call dsbsgammafull(susysratio,1)
deltaratio = susysratio - smratio
699 write(18,*) ' SM BR(b->s_gamma) = ', smratio
write(18,*) ' Total BR(b->s_gamma) = ', susysratio
write(18,*) ' DeltaBR(b->s_gamma) = ', deltaratio

704 c-----limit on dark matter direct search
c... XENON10 experiment, arXiv:0706.0039v1, astro-ph
      call dsddneunuc(sigmap, sigman, otherp, othern)
      sigmaxenon = (sigmap*54.0+sigman*77.0)/131.0
      xenondat = log(mass(kn(1)))/log(10.0)
709 call darkmatterxenon2(xenondat, testxenon)
      sigmaxenon=log(sigmaxenon)/log(10.0)
      write(18,*) ' cross section calculated = ', sigmaxenon
      write(18,*) ' data = ', testxenon

714 c oh2=dsrdomega(1,1,xf,ierr,iwar,nfc)
c oh2prime = dsrdomega(0,1,xf,ierr,iwar,nfc)
c write(*,*) ' Relic abundance without co = ', oh2prime
c write(*,*) ' Relic abundance = ', oh2

719 end if

      close(18)
      END

724 subroutine darkmatterxenon2(x, res)
      real*8 x, res
      res= 35.787621-226.638876*x + 278.145916*x**2 - 192.145996*x**3+
&      81.962485*x**4 -22.088651*x**5 + 3.671239*x**6-
729 &      0.343998*x**7+ 0.01391*x**8
      return
      end

```

Listing 6 – my-code.F- interface IsaSUSY/Micromegas

```

C   this code tests Micromegas functions on relic density
C   deltatho , b-> s gamma, etc..
3
C   By default , the mass loop corrections are calculated by SuSpect
C   To use another package   change 'define' setting below and
C   set path to the correponding package via lib/Makefile

8

#define EwsbMSSM    suspectEwsbMSSM
C#define EwsbMSSM    isajetEwsbMSSM
C#define EwsbMSSM    sphenoEwsbMSSM
13 C#define EwsbMSSM    softSusyEwsbMSSM

    implicit none
C   ! declaration of external functions

18 #include " ../sources/micromegas.fh"
#include " lib/pmodel.fh"

*   external iargc
*   integer iargc
23 C   ! local parameters

    integer    fast/1/
    real*8 Beps/1.E-6/, cut/0.01/
    real*8 Omega, Xf, omg
28    character*20 mess
    character*50 argv
    integer i, err, argc
C   real*8 xmg, xmu, xmh, xtanb, xmq1, xmdr, xmur, amtmp,
C   & xml1, xmer, xmq2, xmsr, xmer, xml2, xmmr, xmq3,
33 C   & xmbr, xmtr, xml3, xmlr, xat, xab, xal, xm1, xm2, xmt

    open(unit=15, file='my_output.txt')

38 C   init of micromegas

C   call load_par(20, xmg, xmu, xmh, xtanb, xmq1, xmdr, xmur, amtmp,
C   & xml1, xmer, xmq2, xmsr, xmer, xml2, xmmr, xmq3,
C   & xmbr, xmtr, xml3, xmlr, xat, xab, xal, xm1, xm2, xmt)

43 C   call assignValW('Am', dble(xal))

C   err=readVar('essai.dat')

48    call isajet2micromegas()
err=0

```



```

C      err= EwsbMSSM()
      if(err<0) then
53         write(*,*) 'Problem with spectrum calculation'
           goto 13
      endif

      err=sortOddParticles(mess)
58      if(err.ne.0) then
           write(*,*) 'Cannot calculate', mess
      else
           call HiggsMasses(15)      ! 15 is the unit of the output
           call printMasses(15,1)

63
           if(mess.ne.'~o1') then
               write(*,*) '~o1 is not LSP'
           else
               call o1Contents(15)

68
               Omega=darkOmega(Xf, fast, Beps)
               write(15,fmt='(A3,F6.2,A7,1PE10.2)')
               'Xf=', Xf, 'Omega=', Omega
C      >      omg=printChannels(Xf, cut, Beps, 1, 6)

73
           write(15,fmt='(A,3x,1PE8.2)') 'deltarho=', deltarho()
           write(15,fmt='(A,3x,1PE8.2)') 'gmuon=', gmuon()
           write(15,fmt='(A,3x,1PE8.2)') 'bsgnlo=', bsgnlo()
           write(15,fmt='(A,3x,1PE8.2)') 'bsmumu=', bsmumu()
78
           if(MassLimits().eq.0) write(*,*) 'MassLimits OK'

           write(15,*) findValW("MH3"), 'Mass of CP-odd Higgs'
           write(15,*) findValW("Mh"), 'h0'
           write(15,*) findValW("MHH"), 'Mass of H Higgs'
83
           write(15,*) findValW("MHc"), 'Mass of charged Higgs'
           write(15,*) findValW("tb"), 'Tangent beta'
           write(15,*) findValW("mu"), 'mu parameter'
           write(15,*) findValW("Am"), 'smu soft coupling'
           write(15,*) findValW("Ab"), 'sBottom soft coupling'
88
           write(15,*) findValW("At"), 'sTop soft coupling'
           write(15,*) findValW("MNE1"), 'Neutralino 1 mass'
           write(15,*) findValW("MNE2"), 'Neutralino 2 mass'
           write(15,*) findValW("MNE3"), 'Neutralino 3 mass'
           write(15,*) findValW("MNE4"), 'Neutralino 4 mass'
93
           write(15,*) findValW("MC1"), '1st chargino mass'
           write(15,*) findValW("MC2"), '2nd chargino mass'
           write(15,*) findValW("MSG"), 'mass of gluino'
           write(15,*) findValW("MSne"), 'e-sneutrino mass'
           write(15,*) findValW("MSnm"), 'mu-sneutrino mass'
98
           write(15,*) findValW("MSnl"), 'tau-sneutrino mass'
           write(15,*) findValW("MSEL"), 'light selectron mass'

```

```

write (15,*) findValW ("MSeR"), 'heavy_selectron_mass'
write (15,*) findValW ("MSmL"), 'light_smuon_mass'
write (15,*) findValW ("MSmR"), 'heavy_smuon_mass'
103 write (15,*) findValW ("MSl1"), 'light_stau_mass'
write (15,*) findValW ("MSl2"), 'heavy_stau_mass'
write (15,*) findValW ("MSuL"), 'left_sup_mass'
write (15,*) findValW ("MSuR"), 'right_sup_mass'
write (15,*) findValW ("MSsL"), 'left_sstrange_mass'
108 write (15,*) findValW ("MSsR"), 'right_sstrange_mass'
write (15,*) findValW ("MSt1"), 'light_stop_mass'
write (15,*) findValW ("MSt2"), 'heavy_stop_mass'
write (15,*) findValW ("MSdL"), 'left_sdown_mass'
write (15,*) findValW ("MSdR"), 'right_sdown_mass'
113 write (15,*) findValW ("MScL"), 'left_scharm_mass'
write (15,*) findValW ("MScR"), 'right_scharm_mass'
write (15,*) findValW ("MSb1"), 'light_sbottom_mass'
write (15,*) findValW ("MSb2"), 'heavy_sbottom_mass'
write (15,*) findValW ("Zn11"), 'neutralino_mixing_Zn11'
118 write (15,*) findValW ("Zn21"), 'neutralino_mixing_Zn21'
write (15,*) findValW ("Zn31"), 'neutralino_mixing_Zn31'
write (15,*) findValW ("Zn41"), 'neutralino_mixing_Zn41'
write (15,*) findValW ("Zn12"), 'neutralino_mixing_Zn12'
write (15,*) findValW ("Zn22"), 'neutralino_mixing_Zn22'
123 write (15,*) findValW ("Zn32"), 'neutralino_mixing_Zn32'
write (15,*) findValW ("Zn42"), 'neutralino_mixing_Zn42'
write (15,*) findValW ("Zn13"), 'neutralino_mixing_Zn13'
write (15,*) findValW ("Zn23"), 'neutralino_mixing_Zn23'
write (15,*) findValW ("Zn33"), 'neutralino_mixing_Zn33'
128 write (15,*) findValW ("Zn43"), 'neutralino_mixing_Zn43'
write (15,*) findValW ("Zn14"), 'neutralino_mixing_Zn14'
write (15,*) findValW ("Zn24"), 'neutralino_mixing_Zn24'
write (15,*) findValW ("Zn34"), 'neutralino_mixing_Zn34'
write (15,*) findValW ("Zn44"), 'neutralino_mixing_Zn44'
133 write (15,*) findValW ("Zu11"), 'chargino_mixing_u11'
write (15,*) findValW ("Zu21"), 'chargino_mixing_u21'
write (15,*) findValW ("Zu12"), 'chargino_mixing_u12'
write (15,*) findValW ("Zu22"), 'chargino_mixing_u22'
write (15,*) findValW ("Zv11"), 'chargino_mixing_v11'
138 write (15,*) findValW ("Zv21"), 'chargino_mixing_v12'
write (15,*) findValW ("Zv12"), 'chargino_mixing_v21'
write (15,*) findValW ("Zv22"), 'chargino_mixing_v22'
write (15,*) findValW ("Zl11"), 'mixing_Zl11'
write (15,*) findValW ("Zl21"), 'mixing_Zl21'
143 write (15,*) findValW ("Zl12"), 'mixing_Zl12'
write (15,*) findValW ("Zl22"), 'mixing_Zl22'
write (15,*) findValW ("Zt11"), 'stau_mixing_Zt11'
write (15,*) findValW ("Zt21"), 'stau_mixing_Zt21'
write (15,*) findValW ("Zt12"), 'stau_mixing_Zt12'
148 write (15,*) findValW ("Zt22"), 'stau_mixing_Zt22'
write (15,*) findValW ("Zb11"), 'sbottom_mixing_Zb11'

```

```

write (15,*) findValW("Zb21"), 'sbottom_mixing_Zb21'
write (15,*) findValW("Zb12"), 'sbottom_mixing_Zb12'
write (15,*) findValW("Zb22"), 'sbottom_mixing_Zb22'
153 write (15,*) findValW("MG1"), 'MSSM_M1_(U(1))_soft_parameter'
write (15,*) findValW("MG2"), 'MSSM_M2_(SU(2))_soft_parameter'
write (15,*) findValW("MG3"), 'mass_of_gluino'
write (15,*) findValW("Ml1"),
&      'Left_slepton_doublet_mass_1st_generations'
158 write (15,*) findValW("Ml2"),
&      'Left_slepton_doublet_mass_(2nd_generations)'
write (15,*) findValW("Ml3"),
&      'Left_slepton_doublet_mass_(3rd_generations)'
write (15,*) findValW("Mr1"),
163 &      'Right_slepton_singlet_mass_(1st_generations)'
write (15,*) findValW("Mr2"),
&      'Right_slepton_singlet_mass_(2nd_generations)'
write (15,*) findValW("Mr3"),
&      'Right_slepton_singlet_mass_(3rd_generations)'
168 write (15,*) findValW("Mq1"),
&      'Left_squarks_doublet_mass_(1st_generation)'
write (15,*) findValW("Mq2"),
&      'Left_squarks_doublet_mass_(2nd_generation)'
write (15,*) findValW("Mq3"),
173 &      'Left_squarks_doublet_mass_(3rd_generation)'
write (15,*) findValW("Mu1"),
&      'Right_up_squarks_singlet_mass_(1st_generation)'
write (15,*) findValW("Mu2"),
&      'Right_up_squarks_singlet_mass_(2nd_generation)'
178 write (15,*) findValW("Mu3"),
&      'Right_up_squarks_singlet_mass_(3rd_generation)'
write (15,*) findValW("Md1"),
&      'Right_down_squarks_singlet_mass_(1st_generatio'
183 write (15,*) findValW("Md2"),
&      'Right_down_squarks_singlet_mass_(2nd_generatio'
write (15,*) findValW("Md3"),
&      'Right_down_squarks_singlet_mass_(3rd_generatio'
write (15,*) findValW("wC1"), 'width_of_chargino_1'
write (15,*) findValW("wC2"), 'width_of_chargino_2'
188 write (15,*) findValW("wNE1"), 'width_of_neutralino_1'
write (15,*) findValW("wNE2"), 'width_of_neutralino_2'
write (15,*) findValW("wNE3"), 'width_of_neutralino_3'
write (15,*) findValW("wNE4"), 'width_of_neutralino_4'
write (15,*) findValW("wSG"), 'width_of_gluino'
193 write (15,*) findValW("wSuL"), 'width_of_u-squark_L'
write (15,*) findValW("wSuR"), 'width_of_u-squark_R'
write (15,*) findValW("wScL"), 'width_of_c-squark_1'
write (15,*) findValW("wScR"), 'width_of_c-squark_2'
write (15,*) findValW("wSt1"), 'width_of_t-squark_1'
198 write (15,*) findValW("wSt2"), 'width_of_t-squark_2'
write (15,*) findValW("wSdR"), 'width_of_d-squark_R'

```

```

203   write(15,*) findValW("wSdL"), 'width_of_d-squark_L'
   write(15,*) findValW("wSsL"), 'width_of_s-squark_L'
   write(15,*) findValW("wSsR"), 'width_of_s-squark_R'
208   write(15,*) findValW("wSb1"), 'width_of_b-squark_1'
   write(15,*) findValW("wSb2"), 'width_of_b-squark_2'
   write(15,*) findValW("wt"), 'width_of_top_quark'
   write(15,*) findValW("wZ"), 'Z-boson_width'
213   write(15,*) findValW("wW"), 'W-boson_width'
   write(15,*) findValW("wSne"), 'e-sneutrino_width'
   write(15,*) findValW("wSnm"), 'mu-sneutrino_width'
   write(15,*) findValW("wSnl"), 'tau-sneutrino_width'
   write(15,*) findValW("wSeL"), 'width_of_1st_selectron'
218   write(15,*) findValW("wSeR"), 'width_of_2nd_selectron'
   write(15,*) findValW("wSmL"), 'width_of_1st_smuon'
   write(15,*) findValW("wSmR"), 'width_of_2nd_smuon'
   write(15,*) findValW("wSl1"), 'width_of_1st_stau'
   write(15,*) findValW("wSl2"), 'width_of_2nd_stau'

223   endif
   end if
13   continue

   close(15)
   end

233   C      subroutine load_par (numb, xxmg, xxmu, xxmha, xxtanb, xxmq1,
   C      &      xxmdr, xxmur, amtmp,
238   C      &      xxml1, xxmer, xxmq2, xxmsr, xxmcr, xxml2, xxmmr, xxmq3,
   C      &      xxmbr, xxmtr, xxml3, xxmlr, xxat, xxab, xxal, xxm1, xxm2, xxmt)
   C
   C      implicit none
   C
233   C      integer numb
   C      character other
   C      init
   C      real*8 xxmg, xxmu, xxmha, xxtanb, xxmq1, xxmdr, xxmur,
   C      &      amtmp,
238   C      &      xxml1, xxmer, xxmq2, xxmsr, xxmcr, xxml2, xxmmr, xxmq3,
   C      &      xxmbr, xxmtr, xxml3, xxmlr, xxat, xxab, xxal, xxm1, xxm2, xxmt

   C      open(unit=numb,
   C      &      file = '../interface_dark_isajet/weak_parameter.txt ')
243   C
   C      read(numb,*) other, xxmu
   C      read(numb,*) other, xxm1
   C      read(numb,*) other, xxm2
   C      read(numb,*) other, xxmg
248   C      read(numb,*) other, xxml1
   C      read(numb,*) other, xxml2

```

```

C      read (numb,*) other , xml3
C      read (numb,*) other , xmmer
C      read (numb,*) other , xmmer
253 C      read (numb,*) other , xmlr
C      read (numb,*) other , xmq1
C      read (numb,*) other , xmq2
C      read (numb,*) other , xmq3
C      read (numb,*) other , xmur
258 C      read (numb,*) other , xmcr
C      read (numb,*) other , xmtr
C      read (numb,*) other , xmdr
C      read (numb,*) other , xmrsr
C      read (numb,*) other , xmbr
263 C      read (numb,*) other , xat
C      read (numb,*) other , xab
C      read (numb,*) other , xal
C      read (numb,*) other , xmha
C      read (numb,*) other , xtanb
268 C      read (numb,*) other , amtmp
C      read (numb,*) other , xmt

C      call assignValW ("mu" , xmu)
C      call assignValW ("MG1" , xm1)
273 C      call assignValW ("MG2" , xm2)
C      call assignValW ("MG3" , xm3)
C      call assignValW ("M1" , xml1)
C      call assignValW ("M2" , xml2)
C      call assignValW ("M3" , xml3)
278 C      call assignValW ("Mr1" , xmmer)
C      call assignValW ("Mr2" , xmmer)
C      call assignValW ("Mr3" , xmlr)
C      call assignValW ("Mq1" , xmq1)
C      call assignValW ("Mq2" , xmq2)
283 C      call assignValW ("Mq3" , xmq3)
C      call assignValW ("Mu1" , xmur)
C      call assignValW ("Mu2" , xmcr)
C      call assignValW ("Mu3" , xmtr)
C      call assignValW ("Md1" , xmdr)
288 C      call assignValW ("Md2" , xmrsr)
C      call assignValW ("Md3" , xmbr)
C      call assignValW ("At" , xat)
C      call assignValW ("Ab" , xab)
C      call assignValW ("Al" , xal)
293 C      call assignValW ("MH3" , xmha)
C      call assignValW ("tb" , xtanb)
C      call assignValW ("Am" , amtmp)
C      call assignValW ("Mtp" , xmt)

298 C      close (numb)
C      return

```

*C end subroutine*

```

303  subroutine isajet2micromegas ()
      IMPLICIT NONE

      REAL isa_input

308  character*10 name
      DOUBLE PRECISION findValW
      REAL ZMIXSS(4,4),GAMMAL,GAMMAR,ALFAH,THETAT,
&      THETAB,THETAL,VUQ,VDQ

313  integer sing
      INTEGER I,J
      CHARACTER*6 NAMES(24),PNAMES(32)
      REAL*8 SMC2
      DATA NAMES/'MG3','mu','MH3','tb_Q','Mq1','Md1','Mu1','Ml1','Mr1',
318 &'Mq2','Md2','Mu2','Ml2','Mr2','Mq3','Md3','Mu3','Ml3','Mr3',
&'At','Ab','Al','MG1','MG2'/
      real tb
      DATA PNAMES/'MSG','MSuL','MSuR','MSdL','MSdR','MSsL',
&      'MSsR','MScL','MScR','MSb1','MSb2','MSt1',
323 &'MSt2','MSne','MSnm','MSnl','MSeL','MSeR',
&      'MSmL','MSmR','MSl1','MSl2','MNE1','MNE2',
&      'MNE3','MNE4','MC1','MC2','Mh','MHH',
&      'MH3','MHc'/

328  OPEN(unit=17,
&      file='../../interface_dark_isajet/isa2micro.txt')

      read(17,*) tb
      call assignValW('tb',dble(tb))
333  DO I=1,24
      read(17,*) isa_input
      call assignValW(NAMES(I),dble(isa_input))
      END DO

338  read(17,*) VUQ,VDQ
      call assignValW('vev',dsqrt(2*dble(VUQ)**2+dble(VDQ)**2))
      call assignValW('tb_Q',DBLE(VUQ/VDQ))
      C call assignValW('QSUSY',DBLE(HIGFRZ))
      C call assignValW('g3',DBLE(sqrt(ASMSS*16*ATAN(1))))
343 C call assignValW('gY',DBLE(SQRT(.6)*GSS(1)))
      C call assignValW('g2',DBLE(GSS(2)))
      C call assignValW('mH1.2',DBLE(GSS(13)))
      C call assignValW('mH2.2',DBLE(GSS(14)))
      C call assignValW('Yl',DBLE(GSS(4)))
348 C call assignValW('Yb',DBLE(GSS(5)))
      C call assignValW('Yt',DBLE(GSS(6)))

```

```

C      tb=VUQ/VDQ
C      call assignValW('mA_2', DBLE(B*XISAIN(2)*(1+tb*tb)/tb))

353 DO I=1,32
      read(17,*) isa_input
      call assignValW(PNAMES(I), dble(isa_input))
END DO

358 do i=1,4
      read(17,*) ZMIXSS(i,1),ZMIXSS(i,2),ZMIXSS(i,3),ZMIXSS(i,4)
end do

do i=1,4
363 do j=1,4
      write(name,FMT='(A2,I1,I1)') 'Zn',j,i
      sing=1
      if((i.eq.3).or.(i.eq.4)) sing=-sing
      if((j.eq.2).or.(j.eq.4)) sing=-sing
368 call assignValW(name,DBLE(sing*ZMIXSS(5-i,j)))
      end do
end do

read(17,*) GAMMAL,GAMMAR
373 SMC2=SIGN(1.,TAN(GAMMAL)*TAN(GAMMAR))
read(17,*) ALFAH
read(17,*) thetal
read(17,*) thetat
read(17,*) thetab

378 call assignValW('Zu11',dble(-sin(GAMMAL)))
call assignValW('Zu12',dble(-cos(GAMMAL)))
call assignValW('Zu21',dble(-cos(GAMMAL)*SMC2))
call assignValW('Zu22',dble(sin(GAMMAL)*SMC2))

383 call assignValW('Zv11',dble(-sin(GAMMAR)))
call assignValW('Zv12',dble(-cos(GAMMAR)))
call assignValW('Zv21',dble(-cos(GAMMAR)))
call assignValW('Zv22',dble(sin(GAMMAR)))

388 call assignValW('alpha',dble(-ALFAH))

call assignValW('Zl11',dcos(dble(thetal)))
call assignValW('Zl12',-dsin(dble(thetal)))
393 call assignValW('Zl12',dsin(dble(thetal)))
call assignValW('Zl22',dcos(dble(thetal)))

call assignValW('Zt11',dcos(dble(thetat)))
call assignValW('Zt21',-dsin(dble(thetat)))
398 call assignValW('Zt12',dsin(dble(thetat)))
call assignValW('Zt22',dcos(dble(thetat)))

```

403

```
call assignValW( 'Zb11', dcos( dble( thetab )))  
call assignValW( 'Zb21', -dsin( dble( thetab )))  
call assignValW( 'Zb12', dsin( dble( thetab )))  
call assignValW( 'Zb22', dcos( dble( thetab )))  
  
call assignValW( 'Am', findValW( 'Al' ))  
end
```