

# Détecteur SDHCAL pour le signal $e^+e^- \longrightarrow \nu\nu h$ : Optimisation et Adaptation de l'analyse de données pour le Projet FCC

Alexia HOCINE

Étudiante en M2 Physique SUBA à l'UCBL1

Supervisé par Gérard GRENIER

Maître de Conférence - UCBL1

Enseignant-Chercheur - IP2I, CNRS, IN2P3

Membre des collaborations CALICE, CMS et ILD

Corresponsable du groupe SDHCAL au sein de la collaboration CALICE



Rapport de Stage - Master 2 Physique SUBA  
Université de Claude Bernard Lyon 1

Avril-Juillet 2022

# Préambule

## Remerciements

Je souhaite d'abord remercier Gérard GRENIER pour m'avoir donné la chance de montrer ce que je peux faire. Et aussi pour son encadrement, son accompagnement et son temps.

Je souhaite plus largement remercier mon équipe, Gérard GRENIER, Imad LAKTINEH et Clément DEVANNE, pour l'atmosphère positive, détendue et stimulante.

Et plus largement, les employés de l'IP2I pour leur gentillesse et leur accueil.

## Participation à la *Geek and Japan Touch*

Au cours de mon stage, j'ai participé à l'atelier tenu par l'IP2I à la *Geek and Japon Touch*, organisé par Stéphanie BEAUCERON (IP2I, CNRS, CMS).

Durant ce week-end, avec 2 autres chercheuses (non physiciennes), Florence BOYER et Liliane DE ARAUJO, on a tenu un débat sur le film *Don't look up : Déni Cosmique* de Adam MCKAY, sur la crédibilité du discours scientifiques.

Ensuite sur le stand, j'ai pu expliquer les bases scientifiques et des recherches menées par le CNRS, CMS, et Virgo au près du grand public.

De plus, j'ai aussi animé le stand de l'association ?? dont l'objectif est d'expliquer les principes de base de la gravité en 2D avec un drap tenu.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	La Physique des Collisionneurs	4
1.1.1	Collisionneurs hadroniques	4
1.1.2	Collisionneurs leptoniques	4
1.2	Physique du boson de Higgs	4
1.2.1	Production du boson de Higgs	4
1.2.2	Détecteur	4
1.2.3	Collisions	5
1.3	Présentation & Objectif du Stage	5
<b>2</b>	<b>Programme ILC</b>	<b>6</b>
2.1	Présentation du Projet ILC	6
2.2	Projet numérique : <b>original</b>	6
2.2.1	Données initiales	7
2.2.2	Conversion des fichiers initiaux en fichiers <b>ROOT</b>	7
2.2.3	Analyses des collisions	8
2.3	Projet numérique : <b>ilcsoft</b>	9
2.3.1	Données	9
2.3.2	Programme : <b>processor</b>	10
2.3.3	Programme : <b>analysis</b>	10
<b>3</b>	<b>Programme FCC</b>	<b>11</b>
3.1	Projet FCC	11
3.1.1	Présentation	11
3.2	Développement Numérique	11
3.3	Travail de Stage	11
3.4	Comparaison avec <b>iLCSoft</b>	11
<b>4</b>	<b>Outils Numériques</b>	<b>12</b>
4.1	<b>nnhScript</b>	12
4.2	<b>nnhTest</b>	12
4.2.1	Programmes <b>testXxCompleted.py</b>	12
4.2.2	Programmes <b>testXxSame.py</b>	12
<b>5</b>	<b>Résultats Physiques</b>	<b>13</b>
5.1	Physique du Higgs	13
5.1.1	Section efficace	13
5.1.2	Nombre de Higgs attendu	13
5.1.3	Propriétés à Préciser	13
<b>6</b>	<b>Conclusion</b>	<b>14</b>
6.1	Travail effectué	14
6.2	Résultats attendus	14
6.3	Au delà	14
<b>A</b>	<b>Résumé du travail effectué</b>	<b>15</b>
A.1	Bibliographie	15
A.2	Tutoriels	15
A.3	Code initial	15
A.4	Code final	15

<b>B</b>	<b>Organisation du Projet</b>	<b>16</b>
B.1	Organisation initiale . . . . .	16
B.2	Organisation finale . . . . .	16
B.3	Le dossier NNH_HOME . . . . .	16
<b>C</b>	<b>Fichiers ROOT de sorties du programme processor</b>	<b>17</b>

# Chapitre 1

## Introduction

### 1.1 La Physique des Collisionneurs

Le principe des collisionneurs est simple, on accélère des particules à des énergies cinétiques suffisantes pour provoquer des collisions inélastiques, et ainsi comprendre les interactions fondamentales et les constituants élémentaires de la matière.

On distingue 2 familles de collisionneurs en fonction des particules qui sont utilisés.

#### 1.1.1 Collisionneurs hadroniques

Les collisionneurs hadroniques utilisent des hadrons, qui sont des particules complexes composées de 3 quarks et de gluons<sup>1</sup>. Par exemple au LHC<sup>2</sup>, on utilise des protons, composés de 2 quarks up et d'1 quark down.

Comme il s'agit de particules composites, ce sont pas les protons qui collisionnent directement mais ces constituants, appelés partons. Chacun porte une fraction indéterminée de l'énergie du proton. On ignore donc l'énergie de la collision en amont, il s'agit d'un paramètre libre.

C'est pourquoi, ils sont utiles pour la découverte de nouvelles particules de masse inconnue, puisqu'ils permettent de balayer tout le spectre de masse sous la gamme d'énergie du collisionneur (au LHC  $< 14 \text{ TeV}$ )<sup>3</sup>.

#### 1.1.2 Collisionneurs leptoniques

En revanche, les collisionneurs leptoniques utilisent des leptons, qui sont des particules élémentaires. Comme le LEP<sup>4</sup>, qui collisionnait des électrons et des positrons [3].

Cette fois-ci, chaque lepton qui collisionne, possède une énergie complète donc connue. Puisqu'on leur impulse une énergie précise, ainsi on augmente la statistique pour un certain niveau d'énergie. Ces collisionneurs sont donc utilisés pour la recherche de précision.

Les prochaines générations de collisionneurs, comme ILC, CEPC, CLIC et FCC, ce sont des collisionneurs leptoniques. Leur objectif est de préciser les données déjà obtenues, notamment sur le boson de Higgs découvert en 2012 par le LHC<sup>5</sup>.

### 1.2 Physique du boson de Higgs

#### 1.2.1 Production du boson de Higgs

Concrètement, on ne mesure pas directement le boson de Higgs mais ses produits de désintégrations sous la forme de jet. Ainsi on cherche à améliorer la résolutions en énergie de ces jets que l'on détecte [6].

#### 1.2.2 Détecteur

En physique des particules, on utilise des détecteurs appelés calorimètres pour mesurer l'énergie des particules. Cette énergie va être déposer par ionisation dans le matériau le long de la trajectoire des particules qui

---

1. Gluon : boson médiateur de l'interaction forte qui maintiennent les quarks ensembles.

2. LHC : Large Hadron Collider, CERN

3. D'où l'intérêt de nouveaux collisionneurs à des énergies plus élevées et donc des masses de particules produites plus lourdes.

4. LEP : Large Electron-Positron, le prédécesseur du LHC, même tunnel

5. Higgs : était la pièce manquante du modèle standard des particules, car il permet aux particules d'acquérir une masse

le traverse. Il faut donc des algorithmes de reconstruction pour déduire les énergies, les types de particules et les trajectoires.

Pour cela, on utilise des calorimètres à grande granularité qui permet une très bonne performance des Algorithmes de Flux de Particules (PFA) [6].

C'est dans ce cadre que la collaboration internationale CALICE, à développer le premier prototype de la famille de calorimètre granulaire SDHCAL, pour Semi-Digital Hadronic CALorimeter, qui a été développé en grande partie à l'IP2I dans l'équipe CMS, auquel j'appartiens pour ce stage.

### 1.2.3 Collisions

Au cours, de ce stage, je me suis concentrée sur les collisions de type **nnh** pour neutrino-neutrino-higgs. Dont voici les diagrammes de Feynman :

FIGURE 1.1 – Diagramme de Feynmann de collision  $e^+e^- \rightarrow \nu\nu h$

## 1.3 Présentation & Objectif du Stage

Pour ce stage, j'ai récupéré les codes de Guillaume GARILLOT, qui les a développés en 2021 au cours de son post-doctorat à l'IP2I. Ils sont en libre accès à l'adresse <https://github.com/ggarillot/nnhAnalysis/tree/refactor>.

Ce programme **nnhAnalysis** permet l'étude de fichiers **SLCIO** pour la collision :

$$e^-e^+ \rightarrow \nu\nu h \quad (1.1)$$

Et l'analyse des canaux de désintégration :

$$h \rightarrow WW^* \rightarrow qq\bar{q}\bar{q} \quad (1.2)$$

$$h \rightarrow b\bar{b} \quad (1.3)$$

Pour cela, il a utilisé les suites logicielles de **iLCSoft**, <https://github.com/iLCSoft> (plus précisément **LCIO** et **Marlin**), qui sont les anciennes suites logicielles. Mais les nouveaux projets de collisionneurs utiliseront **Key4HEP** et **Gaudi**.

Mon objectif est double. Dans un premier temps, comprendre et optimiser les codes existants, c'est-à-dire le programme **nnhAnalysis** de Guillaume GARILLOT qui utilise **LCIO**, **Marlin**. Puis, je vais transformer son programme pour qu'il puisse correspondre aux nouvelles normes des collisionneurs leptoniques, **Key4HEP** et **Gaudi**.

# Chapitre 2

## Programme ILC

### 2.1 Présentation du Projet ILC

Le projet ILC (International Linear Collider) est un collisionneur linéaire, électron-positron, de 31 km conçu pour atteindre une énergie de centre de masse de 500 GeV[2].

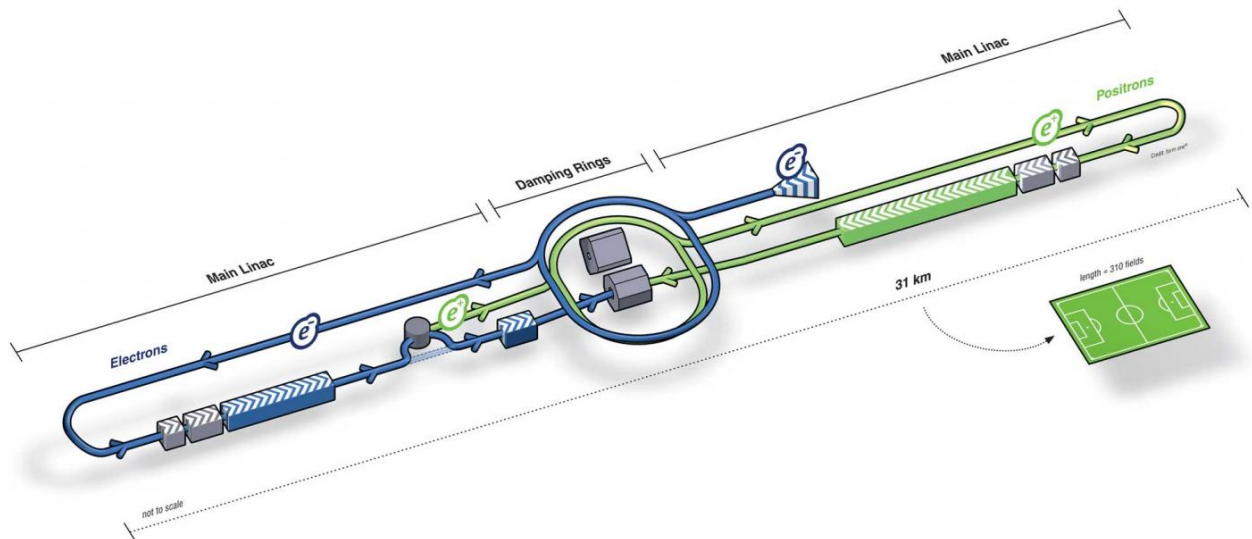


FIGURE 2.1 – Schéma ILC[2]

L'objectif de l'ILC est de produire beaucoup de boson de Higgs, notamment pour découvrir s'il y en a d'autres générations du boson de Higgs. Et plus globalement pour rechercher de la nouvelle physique, par de nouveaux écarts avec le Modèle Standard.

À l'heure actuelle, ce projet attend les autorisations pour lancer sa construction, probablement dans les montagnes au nord du Japon. Et le détecteur SDHCAL correspond parfaitement aux spécifications nécessaires pour un accélérateur linéaire de ce type. C'est pourquoi, afin de consolider sa candidature pour l'appel d'offre, l'IP2I développe aussi des programmes d'analyse de ce détecteur.

### 2.2 Projet numérique : original

Au début de mon stage, j'ai récupéré les codes de Guillaume GARILLOT, qui les a développés en 2021 au cours de son post-doctorat à l'IP2I.

Son projet se divise en 3 programmes indépendants [FIGURE 2.2] :

**miniDSTMarker** qui permet de récupérer les données<sup>1</sup> sur le serveur distant où elles sont stockées.

1. aujourd'hui simulées mais plus tard obtenues dans le détecteur

**processor** permet de tirer, des données brutes précédentes, des arbres ROOT s.

**analysis** utilise les méthodes des arbres binaires boostés pour effectuer une analyse statistique des arbres ROOT issus du programme **processor**.

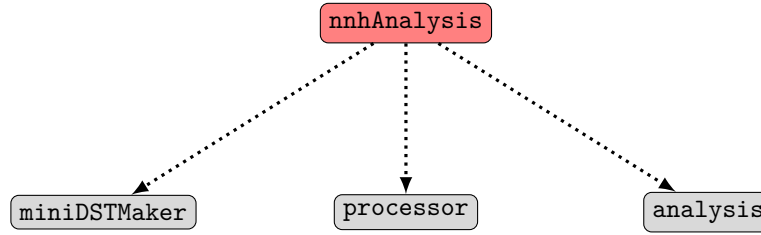


FIGURE 2.2 – Organisation des dossiers de mon Projet - <https://github.com/ggarillot/nnhAnalysis/tree/refactor>

En résumer, ce projet permet l'analyse des collisions survenues dans ce détecteur pour le projet ILC.

### 2.2.1 Données initiales

Malheureusement, pour le temps de ce stage, je n'ai pas obtenu l'accès au serveur, donc je n'ai pas pu utilisé **miniDSTMarker**. Je l'ai donc pas testé, ni pu développer ma propre version. C'est pourquoi, j'ai décidé de ne pas l'inclure dans mon propre code.

Pour que je puisse travailler, on a mis à ma disposition certains de quelqu'un de ces fichiers, ceux des collisions à 250 GeV. Soient 66 dossiers [FIGURE 2.3] de fichiers LCIO, où chaque nom correspond au code du type de processus [FIGURE ??].

```

/gridgroup/ilc/nnhAnalysisFiles/AHCA
(base) [ /gridgroup/ilc/nnhAnalysisFiles/AHCA ]$ ls
402001 402007 402013 500006 500066 500078 500090 500101 500107 500115 500122
402002 402008 402014 500008 500068 500080 500092 500102 500108 500116 500124
402003 402009 402173 500010 500070 500082 500094 500103 500110 500117 500125
402004 402010 402176 500012 500072 500084 500096 500104 500112 500118 500126
402005 402011 402182 500062 500074 500086 500098 500105 500113 500119 500127
402006 402012 402185 500064 500076 500088 500100 500106 500114 500120 500128

```

FIGURE 2.3 – Les noms des dossiers correspondent aux numéros de processus

Type de processus	Code des processus
2 leptoniques	500006, 500008
2 hadroniques	500010, 500012
4 hadroniques	500062, 500064, 500066, 500068, 500070, 500072
4 semi-leptoniques	500074, 500076, 500078, 500080, 500082, 500084, 500101, 500102, 500103, 500104, 500105, 500106, 500107, 500108, 500110, 500112
4 leptoniques	500086, 500088, 500090, 500092, 500094, 500096, 500098, 500100, 500113, 500114, 500115, 500116, 500117, 500118, 500119, 500120, 500122, 500124, 500125, 500126, 500127, 500128
signal	402007, 402173, 402176
autres higgs	402001, 402002, 402003, 402004, 402005, 402006, 402008, 402009, 402010, 402011, 402012, 402013, 402014, 402182, 402185, 402173, 402176

FIGURE 2.4 – Signification des codes des processus, si le signal recherché est de type  $b\bar{b}$ , si le signal est  $WW^*$

### 2.2.2 Conversion des fichiers initiaux en fichiers ROOT

Grâce au programme **processor** on va pouvoir convertir les fichiers initiaux SLCIO en fichiers ROOT standards, afin de pouvoir les analyser.

On obtient ainsi pour chaque dossier de fichier de données SLCIO un fichier ROOT en sortie, c'est-à-dire que l'on obtiendra un arbre ROOT par type de processus.



Ce programme doit être robuste et donc à partir des mêmes fichiers d’entrées toujours générer des fichiers ROOT strictement identiques.

### 2.2.3 Analyses des collisions

#### Étape 1

Avant de commencer l’analyse des fichiers ROOT générés précédemment, on va terminer ce que le programme `processor` avait commencé, et fusionner l’intégralité de ces fichiers en un seul gros fichier `DATA.root`, grâce à la commande `hadd`. Cette commande a été développée par le CERN et elle fusionne tous les histogrammes de différents fichiers en un seul.

Donc là encore, la répétition de ce programme doit toujours générer des fichiers `DATA.root` strictement identique.

#### Étape 2

Pour la deuxième étape de ce programme `analysis`, on va trier nos données en 4 fichiers distincts. Mais au lieu de les séparer par leur numéro de processus, qui est un critère numérique donc éloigné de la réalité des résultats d’une véritable expérience, on va les diviser par le type de particules produit par le boson de Higgs, soit  $b\bar{b}$ , soit  $WW^*$ , et par la polarisation de particules incidentes, c’est-à-dire l’électron et le positron avec une polarisation de -0,8 et 0,3 soit nulles. Ce qui va nous créer les 4 fichiers suivants :

Polarisation	Canal	
$(e, p)$	$b\bar{b}$	$WW^*$
(0, 0)	<code>split_bb_e+0_p+0.root</code>	<code>split_ww_e+0_p+0.root</code>
(-0.8, +0.3)	<code>split_bb_e-0.8_p+0.3.root</code>	<code>split_ww_e-0.8_p+0.3.root</code>

FIGURE 2.5 – Fichiers où les événements sont séparés en fonction des polarisations des particules incidentes et des types de canaux.

Dans chacun de ces fichiers les arbres ROOT, `TTree`, auront les variables suivantes :

**isSignal** : booléen qui indique si on considère l’évènement comme du signal ou du bruit de fond.

**channelType** : entier qui représente le type de canaux de l’évènement [FIGURE 2.4] :

- 2 fermions leptoniques ou hadronique
- 4 fermions leptoniques ou semi-leptonique ou hadroniques
- une autre type de résultat avec un boson de Higgs

**isTrain** booléen qui confirme si l’évènement a été entraîné ou testé par la BDT.

**preSelected** booléen, si l’évènement a été pré-sélectionné par la BDT.

**weight** flottant qui est le poids de l’évènement en  $fb^{-1}$  de la luminosité intégrée.

Pour trier les données du fichier `DATA.root`, on va utiliser des BDT pour *Boosted Decision Tree*, en français, des arbres de décision boostés. Les arbres sont une structure de données courante en informatique, ils sont couramment utilisées pour trier les données car ils ont une rapidité optimale en temps.

Ici les arbres sont utilisés pour déterminer la nature des particules en répondant à des questions booléennes (FIGURE 2.6). Comme il n’y a que 2 réponses pour chaque nœud, on parle d’arbre binaire.

Une fois encore, nous allons utiliser l’API du CERN. En effet, il existe déjà des méthodes (expliquées avec des tutoriels) [1]. **Ces méthodes utilisent des paramètres générées aléatoirement afin d’augmenter la rapidité des calculs. C’est pourquoi on parle d’arbre boosté**

Mais cette fois-ci les fichiers créés sont équivalents mais pas identique. Car les BDT utilisent la génération de nombre aléatoire, ce qui engendre des variations dans les entraînements.

#### Étape 3

Et pour terminer, on peut effectuer l’analyse à proprement parlé de nos données à partir des fichiers `split_XX.root` (FIGURE ??).

Et même si cette analyse sera toujours la même quelque soit les fichiers `split_XX.root`, ces derniers étant légèrement différent d’un entraînement à l’autre les résultats statistiques seront, là encore, légèrement différent mais doivent rester équivalents.

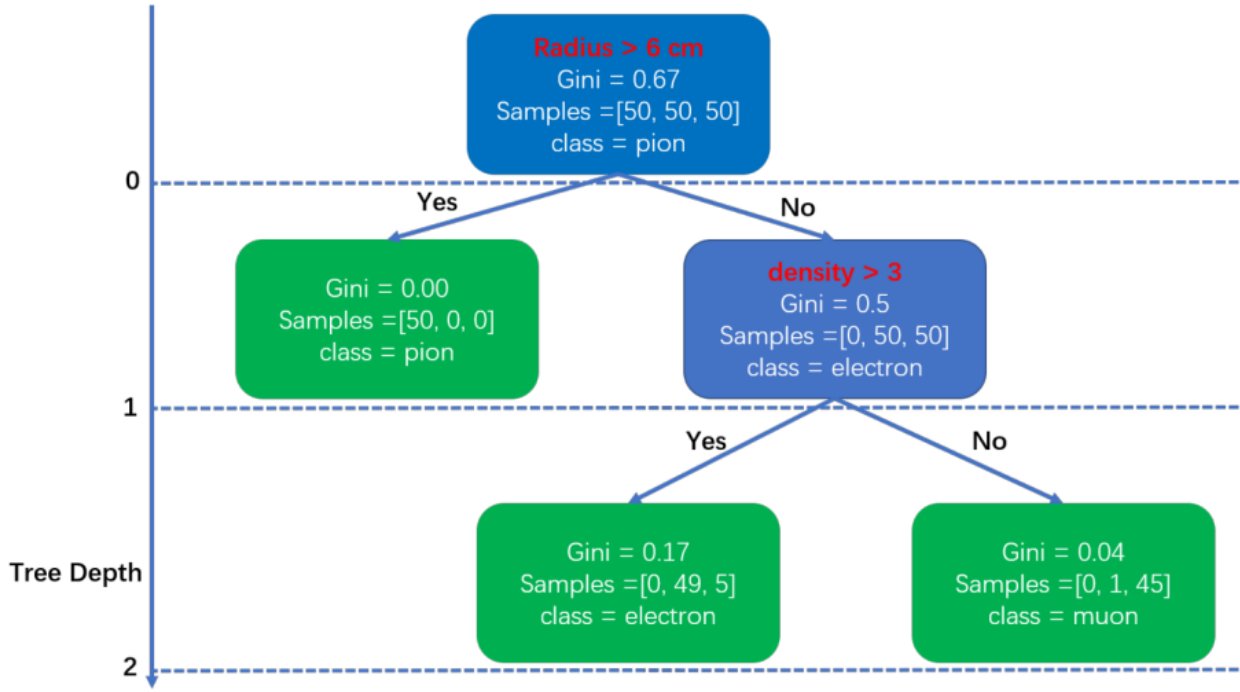


FIGURE 2.6 – Exemple de BDT [6]. Ici, elle permet de déterminer de le type d’une particule

Polarisation	Canal	
$(e, p)$	$b\bar{b}$	$WW^*$
$(-0.8, +0.3)$	model_bb_e-0.8_p+0.3.joblib	model_ww_e-0.8_p+0.3.joblib

FIGURE 2.7 – Fichiers du model de la BDT.

Polarisation	Canal	
$(e, p)$	$b\bar{b}$	$WW^*$
$(0, 0)$	scores_bb_e-0.8_p+0.3.root	scores_ww_e-0.8_p+0.3.root

FIGURE 2.8 – Les fichiers scores ont la réponse de la BDT : 2 variables pour chaque évènement, un booléen pour savoir s’il est sélectionné et la valeur retournée par la BDT.

Polarisation	Canal	
$(e, p)$	$b\bar{b}$	$WW^*$
$(-0.8, +0.3)$	bestSelection_bb_e-0.8_p+0.3.root	bestSelection_ww_e-0.8_p+0.3.root

FIGURE 2.9 – Ce fichier ROOT contient tous les arbres TTree sélectionnés par la BDT comme un évènement de nnh

Polarisation	Canal	
$(e, p)$	$b\bar{b}$	$WW^*$
$(-0.8, +0.3)$	stats_bb_e-0.8_p+0.3.joblib	stats_ww_e-0.8_p+0.3.joblib

FIGURE 2.10 – Fichiers des résultats statistiques pour les polarisations des particules incidentes à 0,8 pour l’électron et à 0,3 pour le positron.

## 2.3 Projet numérique : ilcsoft

### 2.3.1 Données

Initialement, on m’a mis à disposition des fichiers SLCIO, qui sera le format de fichier du détecteur. Chaque fichier est rangés dans un des 66 dossiers (Figure 2.12), qui correspond au code du type de processus.

```

/gridgroup/ilc/nnhAnalysisFiles/AHCAL
(base) [redacted AHCAL]$ ls
402001 402007 402013 500006 500066 500078 500090 500101 500107 500115 500122
402002 402008 402014 500008 500068 500080 500092 500102 500108 500116 500124
402003 402009 402173 500010 500070 500082 500094 500103 500110 500117 500125
402004 402010 402176 500012 500072 500084 500096 500104 500112 500118 500126
402005 402011 402182 500062 500074 500086 500098 500105 500113 500119 500127
402006 402012 402185 500064 500076 500088 500100 500106 500114 500120 500128

```

FIGURE 2.11 – Les noms des dossiers qui correspondent aux numéros de processus

### 2.3.2 Programme : processor

#### Méthodes

On cherche à convertir ces fichiers SLCIO en arbre ROOT par processus.

#### Résultats

Chaque dossier de fichier de donnée SLCIO produira un fichier ROOT en sortie, c'est-à-dire que l'on obtiendra un arbre ROOT par processus.

#### Interprétation

### 2.3.3 Programme : analysis

#### Données

On récupère les fichiers ROOT du programme `processor` précédent.  
*hadd* qui va créer le fichier DATA.root

#### Méthodes

BDT Entraînement

#### L'analyse

#### Résultats

**Vérification des résultats** Comparaison entre les différentes séries d'analyse, basée sur les mêmes fichiers ROOT, mais un autre entraînement de BDT.

#### Interprétation

## Chapitre 3

# Programme FCC

### 3.1 Projet FCC

#### 3.1.1 Présentation

Le FCC (Futur Collisionneur Circulaire) est le projet du CERN pour remplacer leur collisionneur actuelle, le LHC (Large Hadronic Collider). Dont la fin de l'exploitation est prévu en 2040 [4]

Pour le FCC, on prévoit un anneau de 100 km, contre 27 km pour le LEP et le LHC (comme montrer Figure 3.1). Ce qui devrait nous permettra d'atteindre une énergie de 100 TeV contre 13 TeV actuellement pour le LHC.

L'objectif est de rechercher d'une nouvelle physique, en mettant au jour de déviation avec le modèle standard.

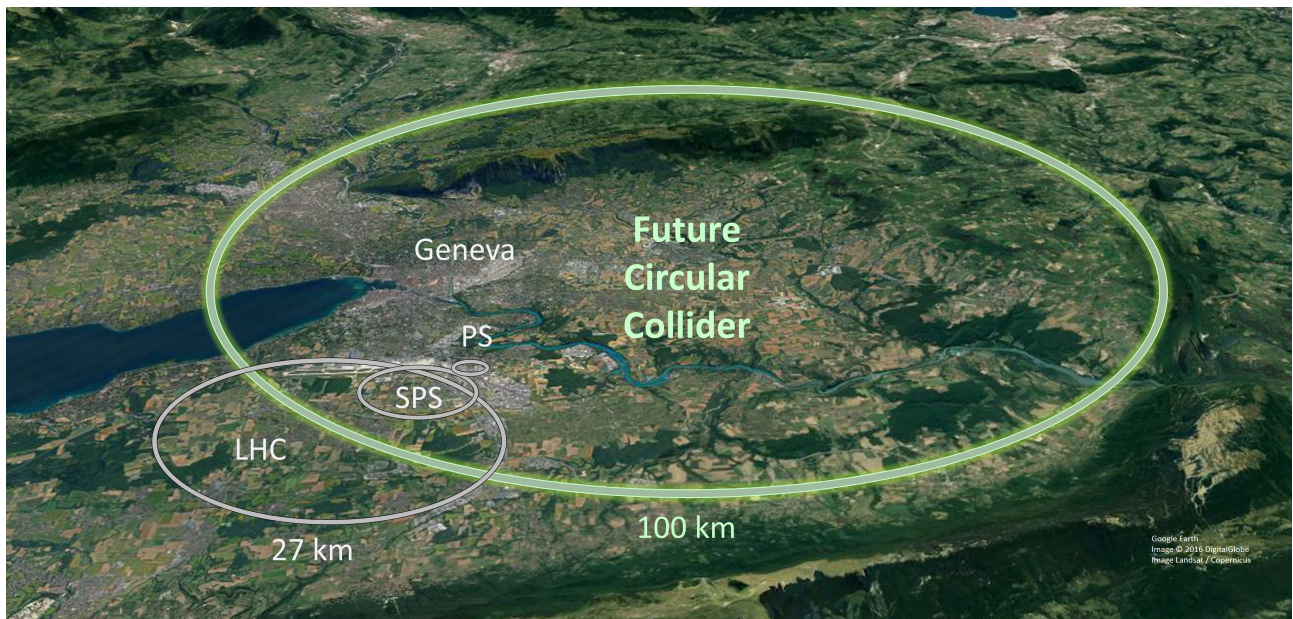


FIGURE 3.1 – <https://cds.cern.ch/images/OPEN-PHO-ACCEL-2019-001-2>

### 3.2 Développement Numérique

Mon objectif dans ce stage est de transformer les codes développés par Guillaume GARILLOT lors de son post-doctorat pour le projet ILC pour ce projet qui n'utilise pas les mêmes suites logiciels.

Gaudi  
EDM4hep

### 3.3 Travail de Stage

### 3.4 Comparaison avec iLCSoft

# Chapitre 4

## Outils Numériques

### 4.1 nnhScript

<https://github.com/alexhxia/nnhAnalysis/tree/main/nnhScript>

### 4.2 nnhTest

Pour tester les programmes générés avec `nnhProgram`, j'ai développé 4 programmes en python :

•	Processus	Analysis
Completed	<code>testProcessorCompleted.py</code>	<code>testAnalysisCompleted.py</code>
Same	<code>testProcessorSame.py</code>	<code>testAnalysisSame.py</code>

FIGURE 4.1 – Tableau récapitulatif des fonctions de tests

#### 4.2.1 Programmes `testXxCompleted.py`

L'objectif de ce type de programme est de tester si tous les fichiers ont été générés.

Programmes `testProcessorCompleted.py`

Le processus est complet si tous les dossiers de

Programmes `testAnalysisCompleted.py`

#### 4.2.2 Programmes `testXxSame.py`

Programmes `testProcessorSame.py`

Programmes `testAnalysisSame.py`

<https://github.com/alexhxia/nnhAnalysis/tree/main/nnhTest>

# Chapitre 5

## Résultats Physiques

### 5.1 Physique du Higgs

#### 5.1.1 Section efficace

**Fusion  $WW^*$**  [5] Comme on est à très haute énergie, on peut approximer que  $\sqrt{s} \gg 2$

$$\sigma_{WW \text{ fusion}} \longrightarrow \frac{g_{HWW}^2 G_F^2}{32 \pi^3} \left[ \left( 1 + \frac{m_H^2}{s} \right) \log \left( \frac{s}{m_H^2} \right) - 2 \left( 1 - \frac{m_H^2}{s} \right) \right] \quad (5.1)$$

$g_{HWW}$  : couplage du boson de Higgs et du boson W

$G_F$  : constante de couplage de Fermi

$m_H$  : masse du boson de Higgs

$s$

#### 5.1.2 Nombre de Higgs attendu

#### 5.1.3 Propriétés à Préciser

## Chapitre 6

# Conclusion

6.1 Travail effectué

6.2 Résultats attendus

6.3 Au delà

# Annexe A

## Résumé du travail effectué

### A.1 Bibliographie

<https://tel.archives-ouvertes.fr/tel-03405418>

- Étude du calorimètre hadronique semi-digital et étude du canal physique

$$e^-e^+ \longrightarrow \nu\nu h \ (H \longrightarrow WW \longrightarrow qqqq)$$

au collisionneur circulaire électron positon (CEPC)

- Bing Liu, IP2I
- 2020

<https://tel.archives-ouvertes.fr/tel-02141420>

- Étude des gerbes hadroniques dans un calorimètre à grande granularité et étude du canal

$$e^-e^+ \longrightarrow HZ \ (Z \longrightarrow qq)$$

dans les futurs collisionneurs leptoniques

- Guillaume Garillot, IPNL
- 2019

### A.2 Tutoriels

LCIO <https://github.com/iLCSoft/LCIO>

ILDConfig <https://github.com/iLCSoft/ILDConfig>

Marlin <https://github.com/iLCSoft/Marlin>

key4hep <https://github.com/key4hep/k4MarlinWrapper>

### A.3 Code initial

<https://github.com/ggarillot/nnhAnalysis/tree/refactor>

### A.4 Code final

<https://github.com/alexhxia/nnhAnalysis>



## Annexe B

# Organisation du Projet

### B.1 Organisation initiale

### B.2 Organisation finale

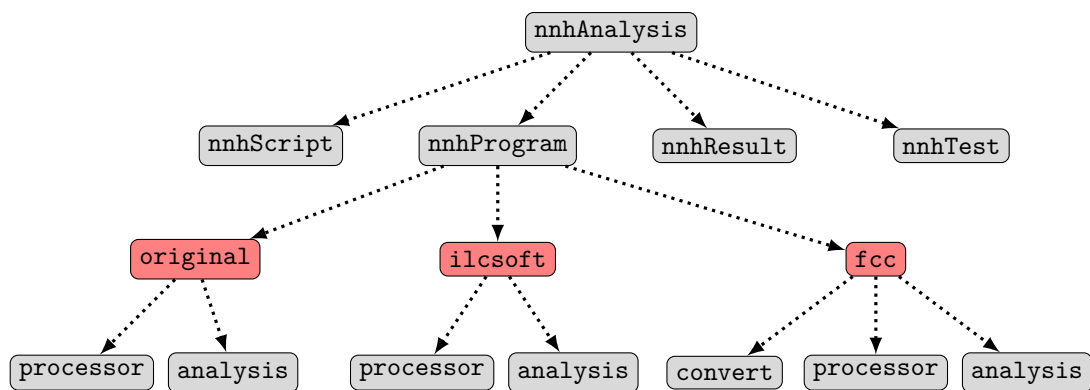


FIGURE B.1 – Organisation des dossiers de mon Projet - <https://github.com/alexhxia/nnhAnalysis>

### B.3 Le dossier NNH\_HOME

Pour s'exécuter, le projet a besoin de la variable d'environnement `NNH_HOME` qui est le chemin du programme que vous souhaitez exécuter, mis en avant en rouge dans les Figure ?? et Figure B.1.

Donc dans le projet initial, il s'agissait de `NNH_HOME=\nnhAnalysis` et dans le nouveau projet :

- `NNH_HOME = \nnhAnalysis\nnhProgram\original`
- `NNH_HOME = \nnhAnalysis\nnhProgram\ilcsoft`
- `NNH_HOME = \nnhAnalysis\nnhProgram\fcc`

## Annexe C

# Fichiers ROOT de sorties du programme processor

On a vu précédemment que `processor` converti une série de fichier `SLCIO` en fichiers `ROOT`.  
Il considère deux canaux de désintégration du Higgs :

$$h \longrightarrow b\bar{b} \tag{C.1}$$

$$h \longrightarrow WW^* \longrightarrow qqqq \tag{C.2}$$

La première équation (C.1) sera mesuré par le détecteur sous la forme de 2 jets reconstruit et identifié comme 2 quark `b`.

Alors que la seconde équation (C.2) sera reconstruit comme 4 jets, 2 par boson `W`<sup>1</sup>.

---

1. `W` sur couche de masse, `Wstar` hors couche de masse

# Table des figures

1.1	Diagramme de Feynmann de collision $e^+e^- \rightarrow \nu\nu h$ . . . . .	5
2.1	Schéma ILC[2] . . . . .	6
2.2	Organisation des dossiers de mon Projet - <a href="https://github.com/ggarillot/nnhAnalysis/tree/refactor">https://github.com/ggarillot/nnhAnalysis/tree/refactor</a> . . . . .	7
2.3	Les noms des dossiers correspondent aux numéros de processus . . . . .	7
2.4	Signification des codes des processus, <b>si le signal recherché est de type <math>b\bar{b}</math>, si le signal est <math>WW^*</math></b> . . . . .	7
2.5	Fichiers où les évènements sont séparés en fonction des polarisations des particules incidentes et des types de canaux. . . . .	8
2.6	Exemple de BDT [6]. Ici, elle permet de déterminer de le type d'une particule . . . . .	9
2.7	Fichiers où les évènements sont séparés en fonction des polarisations des particules incidentes et des types de canaux. . . . .	9
2.8	Fichiers du model de la BDT. . . . .	9
2.9	Les fichiers scores ont la réponse de la BDT : 2 variables pour chaque évènement, un booléen pour savoir s'il est sélectionné et la valeur retournée par la BDT. . . . .	9
2.10	Ce fichier ROOT contient tous les arbres TTree sélectionnés par la BDT comme un évènement de <b>nnh</b> . . . . .	9
2.11	Fichiers des résultats statistiques pour les polarisations des particules incidentes à 0,8 pour l'électron et à 0,3 pour le positron. . . . .	10
2.12	Les noms des dossiers qui correspondent aux numéros de processus . . . . .	10
3.1	<a href="https://cds.cern.ch/images/OPEN-PHO-ACCEL-2019-001-2">https://cds.cern.ch/images/OPEN-PHO-ACCEL-2019-001-2</a> . . . . .	11
4.1	Tableau récapitulatif des fonctions de tests . . . . .	12
B.1	Organisation des dossiers de mon Projet - <a href="https://github.com/alexhxia/nnhAnalysis">https://github.com/alexhxia/nnhAnalysis</a> . . . . .	16

# Bibliographie

- [1] Rene Brun. treefriend.c file reference.
- [2] CERN. International linear collider ready for construction, jun 2013. <https://home.web.cern.ch/news/news/accelerators/international-linear-collider-ready-construction>.
- [3] CERN. The large electron-positron collider, juin 2022. <https://home.cern/science/accelerators/large-electron-positron-collider>.
- [4] CERN. Le futur collisionneur circulaire, jun 2022. <https://home.cern/fr/science/accelerators/future-circular-collider>.
- [5] Klaus Desch and Meyer Niels. Study of higgs boson production through ww-fusion at tesla, Février 2001.
- [6] Bing Liu. *Etude du calorimètre hadronique semi-digital et étude du canal physique  $e^+ e^- \rightarrow H \rightarrow WW \rightarrow qq \bar{q}\bar{q}$  au collisionneur circulaire electron positon (CEPC)*. Theses, Université de Lyon ; Shanghai Jiao Tong University, November 2020.