

Comparaison des prédictions des suites logicielles  
de ILC (iLCSoft) et de FCC (key4HEP)  
sur un signal  $e^+e^- \longrightarrow ZH$

Alexia HOCINE

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 et CMS, ainsi que 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.

## Résumer du travail effectué

Mon stage est principalement du développement numérique, et l'ensemble des codes à conserver par l'IP2I sont sur mon GitHub publique :

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

Email Gérard Grenier :

Un tutorial de ilcsoft : <https://agenda.linearcollider.org/event/9272/>

Initialisation ilcsoft :

La documentation et le packet git du format de données LCIO et de la librairie Marlin

— <https://github.com/iLCSoft/LCIO> [1]

— <https://github.com/iLCSoft/Marlin>

Pour la deuxième partie du stage :

— le software en développement : <https://github.com/key4hep>

— et plus particulièrement l'adaptateur ilcsoft vers key4hep : <https://github.com/key4hep/k4MarlinWrapper>

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Physique des collisionneurs . . . . .	3
1.1.1	Collisionneurs hadroniques . . . . .	3
1.1.2	Collisionneurs leptoniques . . . . .	3
1.2	SDHCAL (Semi-Digital Hadronic CALorimeter) . . . . .	3
1.2.1	Collisions . . . . .	3
1.3	Présentation & Objectif du Stage . . . . .	3
<b>2</b>	<b>ilcsoft</b>	<b>4</b>
2.1	Projet <code>nnhAnalysis</code> . . . . .	4
2.2	Programme <code>processor</code> . . . . .	4
2.3	Programme <code>analysis</code> . . . . .	4
<b>3</b>	<b>FCC</b>	<b>5</b>
3.1	Projet FCC . . . . .	5
3.2	Développement Numérique . . . . .	5
3.3	Travail de Stage . . . . .	5
3.4	Comparaison avec <code>iLCSoft</code> . . . . .	5
<b>4</b>	<b>Outils Numériques</b>	<b>6</b>
4.1	<code>nnhScript</code> . . . . .	6
4.2	<code>nnhTest</code> . . . . .	6
4.2.1	Programmes <code>testXxCompleted.py</code> . . . . .	6
4.2.2	Programmes <code>testXxSame.py</code> . . . . .	6

# Chapitre 1

## Introduction

### 1.1 Physique des collisionneurs

#### 1.1.1 Collisionneurs hadroniques

Collisionneurs pour la découverte

#### 1.1.2 Collisionneurs leptoniques

Collisionneurs de précision

Les prochaines générations de collisionneurs leptoniques sont ILC, CEPC, CLIC et FCC. L'objectif étant une meilleure résolution de l'énergie des jets [2].

Pour cela, on utilise ici des calorimètres à grande granularité qui permet une très bonne performance des Algorithmes de Flux de Particules (PFA) [2].

### 1.2 SDHCAL (Semi-Digital Hadronic CALorimeter)

Dans le cadre de la collaboration internationale CALICE, le premier prototype de la famille de calorimètres granulaires SDHCAL a été développé au grande partie à l'IP2I.

tests en Septembre

#### 1.2.1 Collisions

Au cours, de ce stage, je me concentrerai sur les collisions de type  $\nu\nu h$  pour neutrino-neutrino-higgs

### 1.3 Présentation & Objectif du Stage

analyse des canaux :

$$e^+e^- \longrightarrow \nu\nu h (h \longrightarrow WW \longrightarrow qq\bar{q}\bar{q}) \quad (1.1)$$

$$e^+e^- \longrightarrow \nu\nu h (h \longrightarrow b\bar{b}) \quad (1.2)$$

# Chapitre 2

## ilcsoft

### 2.1 Projet nnhAnalysis

### 2.2 Programme processor

#### Données

Initialement, on m'a mis à disposition des fichiers `SLCIO` rangés par processus dans 66 dossiers (Figure 2.1).

FIGURE 2.1 – Les noms des dossiers qui correspondent aux numéros de processus

Numéro des processus ???

#### Méthodes

On cherche à convertir ces fichiers `SLCIO` en arbre `R00T` par processus.

#### Résultats

Chaque dossier de fichier de donnée `SLCIO` produira un fichier `R00T` en sortie, c'est-à-dire que l'on obtiendra un arbre `R00T` par processus.

#### Interprétation

### 2.3 Programme analysis

#### Données

On récupère les fichiers `R00T` 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 `R00T`, mais un autre entraînement de `BDT`.

#### Interprétation

# Chapitre 3

## FCC

### 3.1 Projet FCC

annaux de 100 km  
photo projet

### 3.2 Développement Numérique

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>

# Table des figures

2.1	Les noms des dossiers qui correspondent aux numéros de processus . . . . .	4
4.1	Tableau récapitulatif des fonctions de tests . . . . .	6