

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.

Table des matières

1	Introduction	3
1.1	Physique des collisionneurs	3
1.1.1	Collisionneurs hadroniques	3
1.1.2	Collisionneurs leptoniques	3
1.2	Physique du boson de Higgs	3
1.2.1	Production du boson de Higgs	3
1.2.2	Détecteur	3
1.2.3	Collisions	4
1.3	Présentation & Objectif du Stage	4
2	ILC	5
2.1	Présentation du Projet ILC	5
2.2	Projet numérique : <code>original</code>	5
2.2.1	Programme : <code>miniDSTMaker</code>	5
2.2.2	Programme : <code>processor</code>	6
2.2.3	Programme <code>analysis</code>	6
2.3	Projet numérique <code>ilcsoft</code>	6
2.3.1	Données	6
2.3.2	Programme : <code>processor</code>	6
2.3.3	Programme <code>analysis</code>	7
3	FCC	8
3.1	Projet FCC	8
3.1.1	Présentation	8
3.2	Développement Numérique	8
3.3	Travail de Stage	9
3.4	Comparaison avec <code>iLCSoft</code>	9
4	Outils Numériques	10
4.1	<code>nnhScript</code>	10
4.2	<code>nnhTest</code>	10
4.2.1	Programmes <code>testXxCompleted.py</code>	10
4.2.2	Programmes <code>testXxSame.py</code>	10
5	Exemples de Résultats Physiques	11
6	Conclusion	12
6.1	Résumer du travail effectué	12
A	Organisation du Projet	13
A.1	Organisation initiale	13
A.2	Organisation finale	13
A.3	Le dossier <code>NNH_HOME</code>	13

Chapitre 1

Introduction

1.1 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 physique.

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 gluons¹. En pratique au LHC (Large Hadron Collider) du CERN, on utilise des protons, 2 quarks up et un quark down.

Comme il s'agit de particules composites, ce sont pas le protons qui collisionnent directement mais ces constituants, appelés partons. Chacun porte une fraction indéterminée de l'énergie du proton. Ce qui permet d'avoir des énergies de collisions inconnues en amont. C'est pourquoi, ils sont utiles pour la découverte de nouvelles particules de masse inconnue, puisqu'il permettent de balayer tout le spectre de masse sous la gamme d'énergie du collisionneur (au LHC < 14 TeV)².

1.1.2 Collisionneurs leptoniques

En revanche, les collisionneurs leptoniques utilisent des leptons, qui sont des particules élémentaires. Comme le LEP (Large Electron-Positron), le prédécesseur du LHC, qui collisionnait des électrons et des positrons [3].

Cette fois-ci, chaque lepton qui collisionne, possède une énergie complète. Donc on connaît parfaitement leur énergie, puisque on peut leur impulser une énergie choisie et ainsi augmenter la statistique pour un niveau d'énergie précis. 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 du LHC, notamment sur le boson de Higgs découvert en 2012 par le LHC, qui était la pièce manquante du modèle standard des particules, car il permet aux particules d'acquérir une masse.

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 les particules qu'il produit sous la forme de jets. Ainsi on veut améliorer la résolutions en énergie de ces jets que l'on détecte [8].

1.2.2 Détecteur

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

1. Gluon : boson médiateurs de l'interaction forte qui maintiennent les quarks ensemble.

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

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) [8].

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 développer 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 concentrerai sur les collisions de type `nnh` pour neutrino-neutrino-higgs. D'où voici les diagrammes de Feynman :

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 :

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

Et l'analyse des canaux de désintégration :

$$h \longrightarrow WW \longrightarrow qq\bar{q}\bar{q} \quad (1.2)$$

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

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 changent de suites logicielles et passent à `Key4HEP` et `Gaudi`.

Mon objectif est double. Dans un premier temps, comprendre et optimiser les codes existants, c'est-à-dire les programmes `LCIO`, `Marlin` et ceux de `nnhAnalysis`. Puis, je vais devoir transformer `nnhAnalysis` pour qu'il puisse correspondre aux nouvelles normes `Key4HEP` et `Gaudi`.

Chapitre 2

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 [1].

L'objectif de l'ILC est de produire beaucoup de boson de Higgs afin de rechercher de la nouvelle physique, par de nouveaux écarts avec le Modèle Standard et il va surtout essayer de découvrir s'il y en a d'autre génération du boson de Higgs.

Ce projet est toujours en attente pour commencer sa construction, probablement dans les montagnes du Nord du Japon. Et le détecteur SDHCAL est en course pour y être installé. C'est pourquoi, l'IP2I développe dans ce programme d'analyse en parallèle du détecteur.

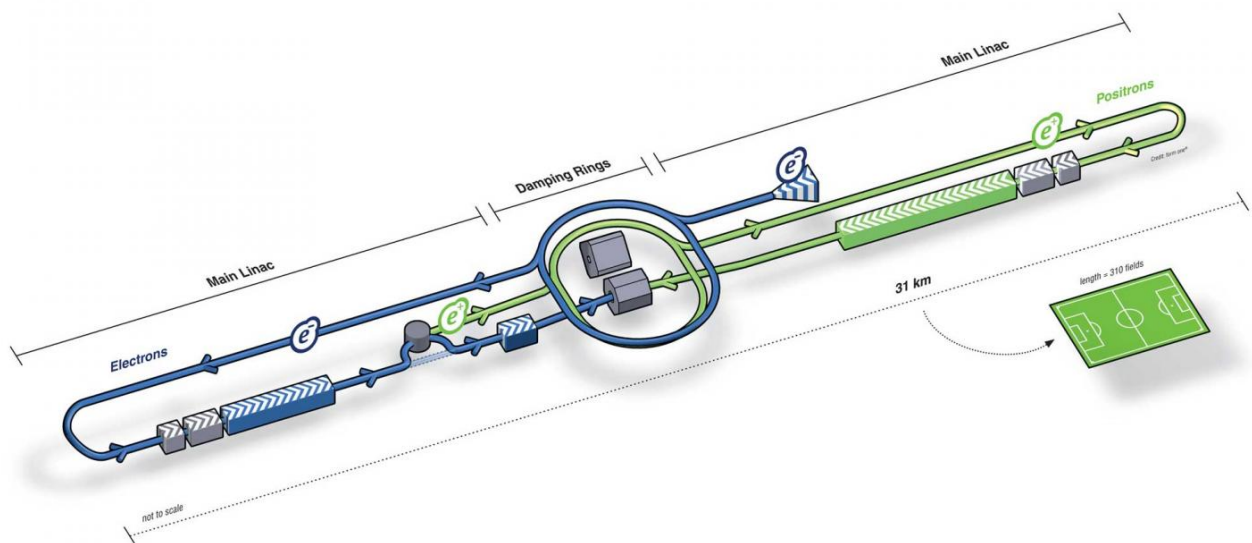


FIGURE 2.1 – Schéma ILC[1]

2.2 Projet numérique : original

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.

2.2.1 Programme : miniDSTMaker

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 2.3), qui correspond au code du type de processus.

```
/gridgroup/ilc/nnhAnalysisFiles/AHCAL
(base) [ ]$ 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 2.2 – Les noms des dossiers qui correspondent aux numéros de processus

2.2.2 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

2.2.3 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

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 2.3), qui correspond au code du type de processus.

```
/gridgroup/ilc/nnhAnalysisFiles/AHCAL
(base) [ ]$ 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 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 `R00T` en sortie, c'est-à-dire que l'on obtiendra un arbre `R00T` par processus.

Interprétation

2.3.3 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

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.

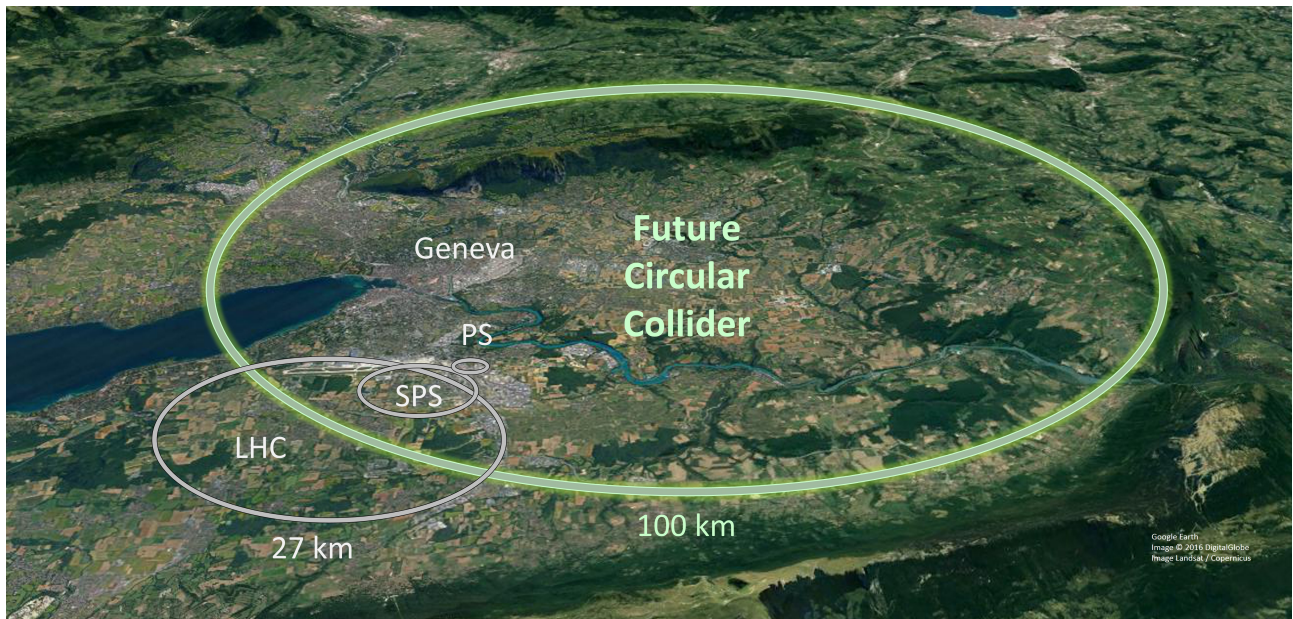


FIGURE 3.1 – <https://cds.cern.ch/images/OPEN-PHO-ACCEL-2019-001-2>

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

Exemples de Résultats Physiques

Chapitre 6

Conclusion

6.1 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 Gérald 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> [6]

— <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>

Annexe A

Organisation du Projet

A.1 Organisation initiale

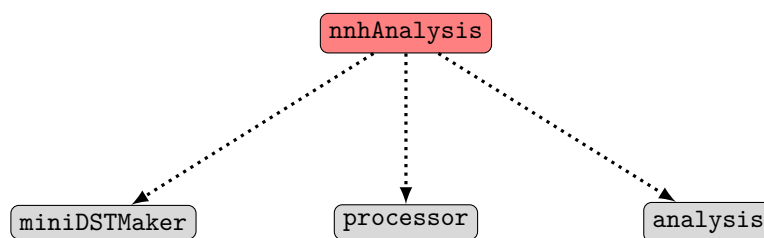


FIGURE A.1 – Organisation des dossiers de mon Projet - <https://github.com/ggarillot/nnhAnalysis/tree/refactor>

A.2 Organisation finale

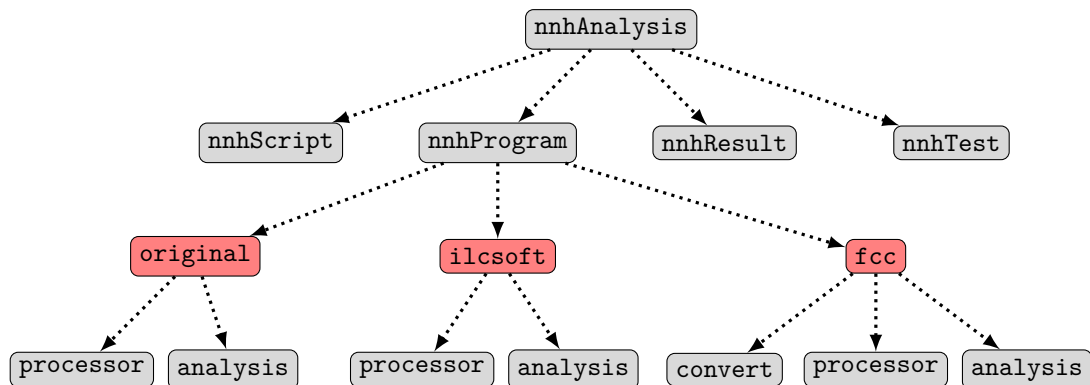


FIGURE A.2 – Organisation des dossiers de mon Projet - <https://github.com/alexhxia/nnhAnalysis>

A.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 A.1 et Figure A.2.

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

Table des figures

2.1	Schéma ILC[1]	5
2.2	Les noms des dossiers qui correspondent aux numéros de processus	6
2.3	Les noms des dossiers qui correspondent aux numéros de processus	6
3.1	https://cds.cern.ch/images/OPEN-PHO-ACCEL-2019-001-2	8
4.1	Tableau récapitulatif des fonctions de tests	10
A.1	Organisation des dossiers de mon Projet - https://github.com/ggarillot/nnhAnalysis/tree/refactor	13
A.2	Organisation des dossiers de mon Projet - https://github.com/alexhxia/nnhAnalysis	13

Bibliographie

- [1] CERN. International linear collider ready for construction, jun 2013. <https://home.web.cern.ch/news/news/accelerators/international-linear-collider-ready-construction>.
- [2] CERN. Grand collisionneur de hadrons, juin 2022. <https://home.cern/fr/science/accelerators/large-hadron-collider>.
- [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] Nicolas Chadeau. Étude sur la mesure de la constante de couplage g_{hw} dans le cadre de l'expérience ilc à 250 gev. Master's thesis, UCBL1, 2021.
- [6] Frank Gaede, Ties Behnke, Norman Graf, and Tony Johnson. Lcio : A persistency framework for linear collider simulation studies. *eConf*, C0303241 :TUKT001, 2003.
- [7] Guillaume Garillot. *Étude des gerbes hadroniques dans un calorimètre à grande granularité et étude du canal $e+e- \rightarrow HZ$ ($Z \rightarrow qq$) dans les futurs collisionneurs leptoniques*. Theses, Université de Lyon, February 2019.
- [8] Bing Liu. *Etude du calorimètre hadronique semi-digital et étude du canal physique $e+e- \rightarrow H \nu \nu$ ($H \rightarrow WW \rightarrow qq qq$) au collisionneur circulaire electron positon (CEPC)*. Theses, Université de Lyon ; Shanghai Jiao Tong University, November 2020.