

Université Catholique de Louvain
Faculté des Sciences
**LPHY2135-Computing et Méthodes numériques en physique des
particules**
Bruno Giacomo Luca

Calorimétrie et boson de Higgs

Maëlle Joveneau
Manuel Tondeur
Brieuc François

Table des matières

1	Introduction	3
2	Descriptifs des méthodes et résultats	3
2.1	Description du détecteur	3
2.2	Description physique du processus initial	4
2.3	Reconstruction de la masse invariante du boson de Higgs	5
2.4	Analyse des résultats	9
3	Conclusion	10

1 Introduction

Afin de découvrir les principes de bases de Geant4, un framework conçu pour la physique des particules, nous avons créé un programme permettant de déterminer la masse du Higgs en suivant la désintégration de celui-ci en deux Z^0 qui eux-mêmes se désintègrent en $e^- e^+$.

Pour ce faire nous avons d'abord dû construire un détecteur qui soit efficace pour notre désintégration. Nous avons déterminé que les éléments nécessaires pour ce détecteur étaient : des calorimètres électromagnétiques et des trackers. Nous avons donc dû utiliser les classes de Geant4 pertinentes pour nos besoins.

Nous avons alors généré nous-même les différentes particules qui apparaissent dans l'évènement qui nous intéresse. Pour donner aux différentes particules leur masse, leur énergie, et/ou leur impulsion, nous avons dû faire appel à des fonctions de distribution variées qui nous ont permises de mettre en pratique la théorie des variables aléatoires dont la méthode "Monte Carlo".

A partir des classes abstraites de Geant4, nous avons défini les interactions possibles entre les particules et les éléments présents au sein du détecteur. Cela fait il nous fallait reconstruire la trajectoire des électrons à partir des informations fournies par les trackers et les calorimètres. Avec celles-ci, nous avons pu déterminer l'énergie et l'impulsion des quatre électrons qui nous donnent la masse du Higgs par la relation de la masse invariante suivante :

$$M_H^2 = E_H^2 - p_H^2 = (E_1 + E_2 + E_3 + E_4)^2 - (\vec{p}_1 + \vec{p}_2 + \vec{p}_3 + \vec{p}_4)^2. \quad (1)$$

Ensuite, les erreurs dues aux bruits ont été prises en compte en posant un seuil d'énergie au niveau de la détection. Pour inclure les erreurs systématiques, 4 électrons d'énergie connues ont été lancés 1200 fois dans le détecteur. Les différences entre l'énergie détectée et l'énergie vraie sont alors mesurées et pondérées avec l'énergie vraie.

Les détails de toutes ces étapes sera présentés dans la suite.

2 Descriptifs des méthodes et résultats

2.1 Description du détecteur

Pour la construction du détecteur nous avons opté pour une géométrie cylindrique de type CMS que nous avons bien entendu simplifiée pour ne garder que les éléments utiles à notre simulation. Le détecteur est donc constitué de plusieurs trackers en forme de tubes imbriqués et d'un calorimètre électromagnétique granulaire qui englobe le tout. Ces éléments sont placés dans le "world volume" que nous avons choisi de remplir d'air pour se rapprocher de la réalité.

Comme dans le CMS, les trackers sont faits de silicium et mesurent 27.2 m de long. Ils sont au nombre de treize, nous en avons mis trois très proches du

centre, le premier étant à 8 cm du vertex et les deux autres espacés de 1 cm du premier. Les dix autres trackers se trouvent espacés chacun de 10 cm, le premier ayant un rayon 20 cm. Bien qu'il nous faille plusieurs trackers pour obtenir une bonne résolution sur la direction des particules, ce choix particulier est fait pour se rapprocher du CMS. Étant donné que notre projet ne nécessite aucunement un champ magnétique, nous avons simplement choisi de ne pas en mettre pour ne pas nous compliquer la tâche inutilement.

Le bloc de calorimètres est un gros tube de 28 m de long dont l'épaisseur s'étend de 1.5 m du centre à 3 m du centre. Il est uniquement composé de calorimètres électromagnétiques car l'événement d'intérêt n'implique ni muon ni hadron. L'épaisseur fût choisie pour éviter toute perte d'énergie.

Les cellules, composées d'Iodure de Cesium, sont relativement grosses par rapport à celles du CMS. En effet, les événements considérés au CMS impliquent un grand nombre de particules qu'il faut pouvoir distinguer les unes des autres. Notre simulation n'ayant que quatre particules identiques à considérer, nous pouvons tout à fait nous permettre cette simplification. La taille des cellules fut donc choisie pour correspondre à la taille d'un jet de sorte que l'énergie d'un jet se dépose presque toujours dans maximum quatre d'entre elles.

NB : on trouvera les détails de la construction des détecteurs dans les classes suivantes : `"projetZTrackerParametrisation.cc"`, `"projetZCaloParametrisation.cc"`, `"projetZTrackerSD.cc"`, `"projetZCaloSD.cc"` et `"projetZDetectroConstruction.cc"`.

2.2 Description physique du processus initial

Pour rappel, le but de ce projet est de reconstruire la masse invariante du boson de Higgs à l'aide de la détection de l'énergie de quatre électrons issus du processus suivant :

$$H \longrightarrow ZZ \longrightarrow e^+e^-e^+e^-. \quad (2)$$

Voyons maintenant en détail la méthodologie utilisée dans ce but. Dans un soucis de réalisme, nous avons choisi de ne pas fixer la masse du H mais de la générer aléatoirement avec une distribution de Breit-Wigner centrée sur 200 GeV. Nous avons dans un premier temps étudié la reconstruction de sa masse invariante lorsque le higgs se désintègre au repos pour ensuite se rapprocher de la réalité et considérer sa désintégration alors qu'il est en mouvement dans le repère du détecteur. Dans ce but, nous générons une énergie cinétique issue d'une distribution exponentielle négative de paramètre 30 GeV ainsi que deux angles θ et φ issus de distributions uniformes pour lui donner une direction¹. NB : la distribution uniforme des angles se justifie par l'absence d'information sur les origines du H . Nous nous plaçons ensuite dans le repère où il est au repos pour simuler sa désintégration en deux bosons Z . La cinématique relativiste du processus est assez simple et tous les paramètres sont fixés hormis la direction d'émission (nous avons choisi

1. On trouvera les différentes distributions dans la classe `"projetZDistribution.cc"`.

de négliger l'incertitude sur la masse du Z car cela est sans intérêt pour notre discussion). Pour la direction, nous générons à nouveau deux angles de manière uniforme mais la justification vient cette fois de l'absence de spin du H .

La même méthode est appliquée pour simuler la désintégration des Z en e^+e^- . Nous avons donc quatre électrons d'énergie et de direction bien déterminées dans le repère de leur parent Z respectif. Il suffit alors d'effectuer les boosts de Lorentz adéquats pour retourner dans le repère du H et ensuite dans le repère du détecteur². On trouvera les détails de ces manipulations dans la classe "*projet-ZElectronsGenerator.cc*". NB : les temps de vie très court du H et du Z nous permettent en très bonne approximation de considérer que ces quatre électrons sont émis au centre géométrique de notre détecteur. L'étape suivante consiste à "lancer" ces quatre électrons dans notre détecteur à l'aide du framework Geant4. Cela est implémenté dans la classe "*projetZPrimaryGeneratorAction.cc*".

Le choix des interactions discrètes autorisées pour chacune des particules est assez simple puisque nous n'avons en jeu que des électrons-positrons et des photons cf "*projetZPhysicsList.cc*". Le voici tout de même :

- photons : création de paires électrons-positrons, diffusion Compton et effet photoélectrique.
- électrons : ionisation, bremsstrahlung et diffusion multiple.
- positrons : ionisation, bremsstrahlung, diffusion multiple et annihilation avec un électron.

2.3 Reconstruction de la masse invariante du boson de Higgs

Une fois les électrons lancés dans le détecteur, nous obtenons une réponse sous forme d'un ensemble de "hits" aussi bien pour le tracker que pour le calorimètre. NB : les détails des hits se trouvent dans les classes "*projetZTrackerHit.cc*" et "*projetZCaloHit.cc*". L'information d'intérêt dépend de la pièce du détecteur eg la position du hit pour les trackers et l'énergie déposée pour le calorimètre.

Présentons les choses dans l'ordre d'utilisation de notre code. Chaque cellule du calorimètre nous renvoie l'énergie totale déposée en elle. Grâce à un algorithme de cluster qui sera explicité plus loin, nous pouvons isoler de cela quatre jets distinct (dans la majorité des cas) et extraire de ces informations l'énergie totale et l'endroit où chacun des électrons s'est arrêté. L'énergie déposée dans les trackers et dans l'air est en effet insignifiante par rapport aux énergies en jeu. Comme nous avons pu éviter de mettre un champ magnétique, la position des différents jets nous donne une information précieuse sur la direction de l'impulsion initiale des électrons aux vertex. Cependant, nos cellules ayant une relativement grande taille, la précision sur la direction est assez médiocre... C'est pourquoi nous avons choisi d'affiner cette information à l'aide des trackers. Avoir une idée de la direction des impulsions grâce aux cellules du calorimètres simplifie grandement la méthode pour obtenir cette même information précisée à l'aide des trackers. En effet, nous

2. Toutes ces manipulations pour simuler la désintégration du higgs sont des algorithmes ne nécessitant pas le framework "Geant4".

avons considéré un cône d'ouverture 40° (valeurs affinées par essais-erreurs) dont la pointe est le vertex et dont l'axe relie ce dernier à la position centrale du jets. La "vraie" direction de l'impulsion initiale de l'électron est obtenue en faisant une moyenne de toutes les directions données par le tracker en ne tenant compte que des hits à l'intérieur de ce cône. NB : l'utilisation du cône est nécessaire car le tracker nous donne les hits de tous les électrons y compris ceux qui rebondissent sur le calorimètre et reviennent dans le tracker.

Pour l'algorithme de cluster, nous avons opté pour une solution assez simple et immédiate. L'algorithme parcourt chacune des cellules et compare l'énergie détectée dans cette cellule avec celles détectées dans les cellules voisines. Si la cellule en question est celle qui possède le maximum, nous la retenons. L'énergie totale du jet est obtenue en y ajoutant les trois énergies maximales voisines. Pour que cet algorithme soit efficace, il faut bien entendu supprimer toutes les cellules ayant reçu peu d'énergie due au du bruit. Après plusieurs essais, nous avons choisi de fixer le seuil à 6 GeV pour éviter de compter parfois 5 jets au lieu de quatre à cause du bruit (le bruit est tiré aléatoirement d'une gaussienne centrée sur 0 et de largeur 1 GeV et il arrivait que des énergies fantômes montent jusqu'à 5 GeV). Bien sûr, cette énergie seuil donne lieu à des erreurs systématiques lorsque le jet est réparti par exemple sur trois cellules et que la quatrième ne reçoit que quelques GeV. Cette énergie est donc éliminée par le seuil, mais ce problème peut-être en partie solutionné. En effet, une fois calibré, la perte moyenne d'énergie due à ce phénomène est corrigée si nous prenons nos données sur un suffisamment grand nombre d'événement. Le seul effet négatif est l'élargissement de la gaussienne centrée sur la "vraie" valeur.

NB : l'algorithme du cluster et de la reconstruction de la direction des impulsions au vertex peut-être trouvé dans "*projetZRunManager.cc*"

Cet algorithme possède un autre défaut, lorsque deux jets sont l'un à côté de l'autre, l'énergie de certaines cases provient des deux jets et notre algorithme la comptera donc deux fois. A titre d'exemple visuel, voici quelques graphiques illustrant les différents cas rencontrés. Sur la figure (1) on voit un événement avec bruit correspondant aux cas où notre algorithme fonctionne parfaitement. Le même cas après imposition du seuil d'énergie est illustré sur la figure (2). Ensuite viennent les cas où nous rencontrons le problème du surcomptage de l'énergie sur les cases "partagées" (voir figure (3) et (4)).

FIGURE 1 – Résultat typique d’une détection de 4 jets parfaits par le cluster avec bruit

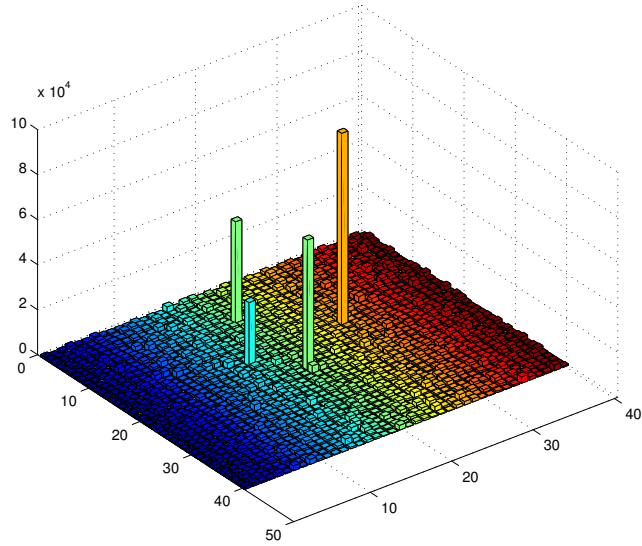


FIGURE 2 – Résultat typique d’une détection de 4 jets parfaits par le cluster après nettoyage du bruit

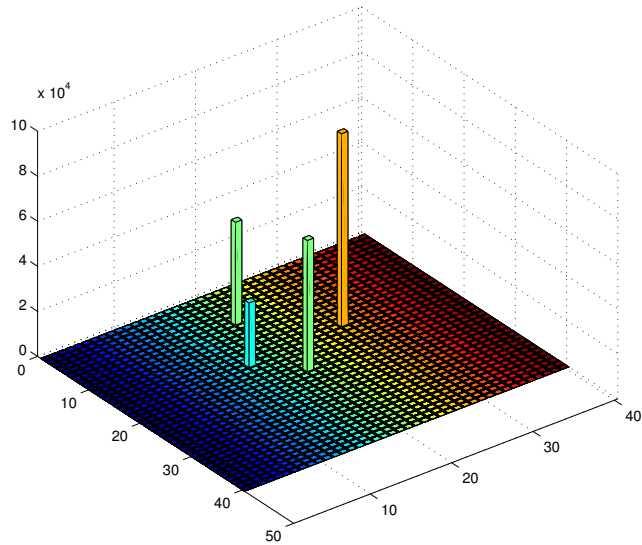


FIGURE 3 – Résultat d’une détection dite à problème.
On peut voir au niveau des deux jets les plus à gauche le problème du double comptage d’énergie

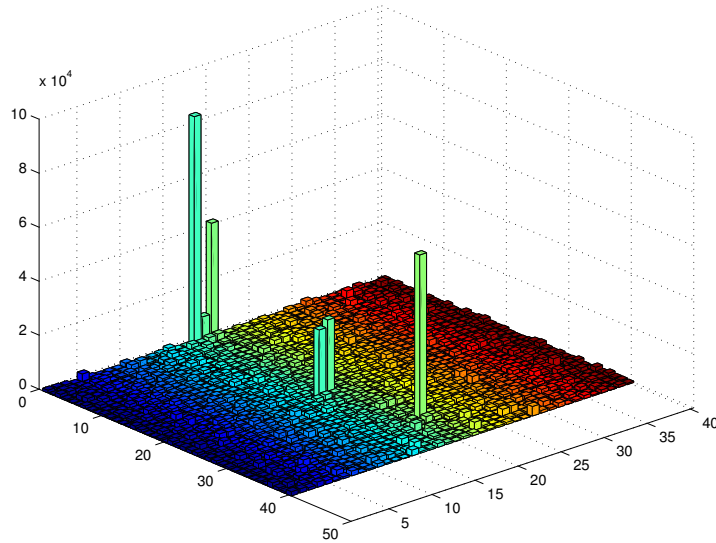
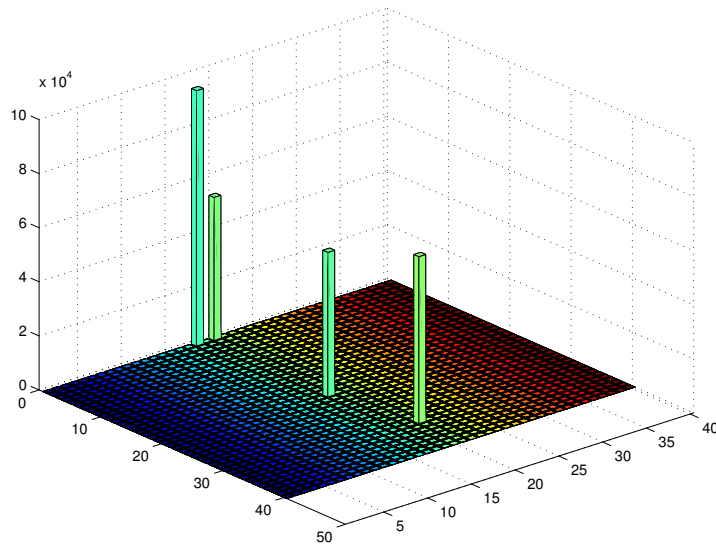


FIGURE 4 – Même après nettoyage du bruit, le problème reste présent



2.4 Analyse des résultats

Une fois tout cela implémenté, nous avons fait tourner le programme en boucle 1500 fois pour mesurer un grand nombre de masse du higgs. C'est avec grand plaisir que nous constatons que la distribution obtenue est fort proche de celle attendue. il suffit de regarder les figures (5) et (6) pour voir que, bien que l'algorithme ne soit pas parfait, les résultats sont plus qu'acceptables. Nous obtenons en effet une moyenne de 199.8950 GeV pour un écart type de 7.0826 GeV. Sur les 1500 higgs créés, 1356 ont générés des événements que notre algorithme à pu identifier correctement et reconstruire. Cela fait moins de 10% de pertes. Une première remarque sur l'histogramme des résultats est que le nombre maximum d'événements se trouve un peu au dessus du 200 GeV mais cela serait explicable par une fluctuation statistique. Une seconde remarque est l'apparition d'événement fort éloignés de 200 GeV probablement dûs aux failles de nos algorithmes. C'est un peu facile de dire ça je sais mais je vois pas trop comment ça peut arriver... Une idée ???

FIGURE 5 – Distribution Breit-Wigner des masses des higgs générés

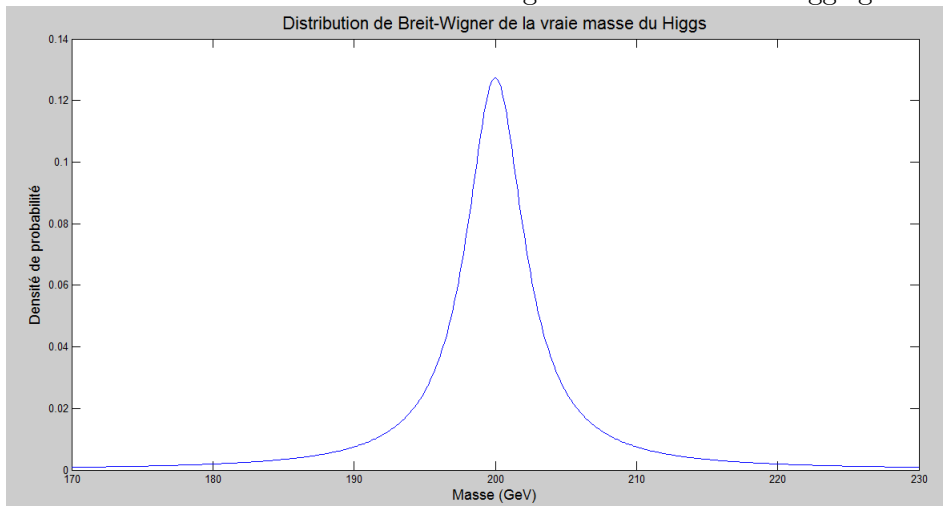
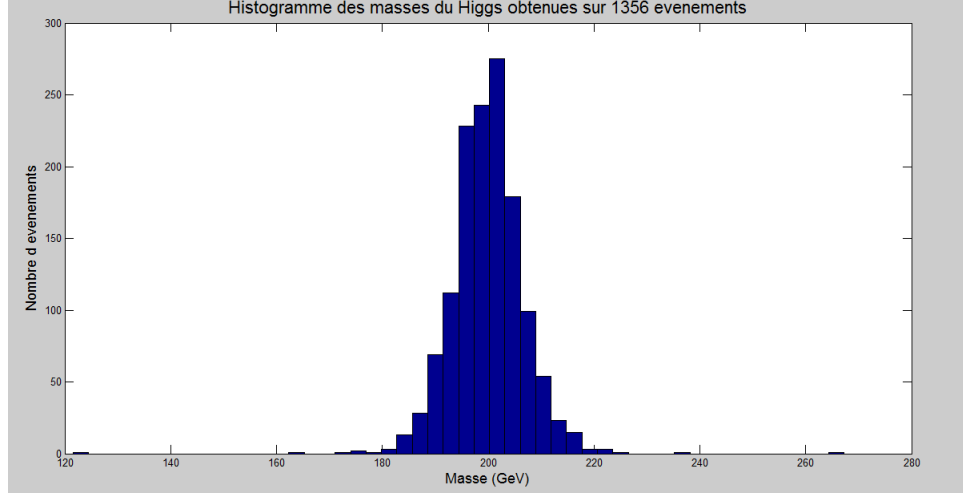


FIGURE 6 – Histogramme des masses mesurées par notre détecteur



3 Conclusion

Il apparait clairement dans la section précédente que nous avons réussi à obtenir la masse du Higgs. Cependant, notre programme n'est pas optimal, il serait possible de le rendre plus rapide et moins couteux en mémoire vive. La précision pourrait également être augmentée.

Une première amélioration serait de refermer le détecteur c'est-à-dire de placer des détecteurs aux extrémités, perpendiculairement à l'axe des z dans notre cas (en laissant évidemment un espace libre pour permettre l'envoi des particules dans le détecteur).

Au niveau de l'algorithme du cluster, on remarque que si deux jets sont très proches l'un de l'autre (une seule case séparant deux maxima d'énergie détectés) alors l'énergie se trouvant sur cette case est comptée deux fois. Cet algorithme pourrait être modifié de sorte que la quantité soit partagée équitablement entre les deux jets. Le plus simple serait alors de se contenter de diviser la quantité par deux mais mieux encore serait de regarder où se trouve le maxima entre les deux pics détectés. En considérant que chaque jet contient toujours la même énergie, la quantité serait répartie en conséquence entre les deux jets (une plus grande partie pour le plus petit des deux maxima et vice-versa).

Une autre façon de limiter le surcomptage de l'énergie décrit dans le paragraphe précédent, serait de raffiner les cellules. En effet, cela diminuerait la probabilité de "case" partagée et même dans les cas où cela se produirait, la quantité d'énergie partagée serait moindre. Cependant cela nécessiterait un algorithme totalement différent étant donné que le notre est basé sur le fait qu'un jet ne puisse pas occuper plus de quatre cellules.

Un dernier point concernant l'algorithme de cluster serait de limiter le nombre de higgs "perdus". En effet, dans le cas présent, nous jetons tous les événements ou nous ne détectons pas exactement quatre jets. Seul un algorithme bien plus

complexe pourrait nous permettre d'exploiter certains de ces cas particuliers. Les higgs étant ici simulés informatiquement, il ne nous coûtent pas très cher.

L'algorithme de reconstruction de la trace est assez basique et par conséquent pas des plus précis. Il aurait sans doute été préférable d'utiliser un algorithme plus complexe du type régression linéaire.

Enfin notons que les cellules de vrais détecteurs sont des cellules focalisées ce qui n'est évidemment pas le cas dans notre simulation, il en résulte des effets indésirables comme des rebonds de particules sur les cellules du calorimètre. Ces effets diminuent également la précision des résultats bien qu'ils ne soient pas présents dans un cas réel.

Certaines de ces améliorations auraient demandées beaucoup de temps à développer pour parfois peu d'amélioration de précision. Nous avons tenté de faire au plus proche de la réalité en passant du temps sur les points qui nous semblaient prioritaires. Nous avons cherché le juste milieu entre l'amélioration des résultats obtenues et le temps demandé pour y parvenir tout en ne négligeant pas nos autres cours et obligations.