Master Science de la matière
École Normale Supérieure de Lyon
Université Claude Bernard Lyon I

Rapport de stage
Samuel Niang
M2 Physique - Concepts et applications

Une méthode de calibration non paramétrique pour les calorimètres de CMS.

Résumé :
Dans le détecteur CMS, l'énergie des hadrons neutres est déterminée à partir de l'énergie mesurée dans les calorimètres électromagnétiques (E_{ecal}) et hadroniques (E_{hcal}). Une calibration est cependant nécessaire pour estimer l'énergie vraie du hadron neutre à partir de E_{ecal} et E_{hcal}. Dans un premier temps, j'ai utilisé comme calibration une fonction linéaire de E_{ecal} et E_{hcal}. Ensuite, afin de décrire la non linéarité de la mesure de l'énergie, j'ai inventé une nouvelle méthode de calibration non paramétrique.

Mots clefs : *Calibration, Modélisation, Physique des particules*

Stage encadré par :
Colin Bernet `colin.bernet@cern.ch`
Bâtiment Paul Dirac
4, Rue Enrico Fermi
69622 Villeurbanne Cedex
Tél. : +33 (0) 4 72 44 84 57

21 juillet 2017

Table des matières

1 Introduction

Après avoir permis la découverte expérimentale du boson de Higgs en 2012, les expériences généralistes ATLAS [1] et CMS [2] installées sur le LHC du CERN, sont toujours en place dans l'optique de découvrir de la nouvelle physique au-delà du modèle standard.

Les détecteurs ATLAS et CMS sont basés sur les mêmes principes : cylindriques, ils sont constitués d'un ensemble de sous-détecteurs disposés en couches concentriques autour du point d'interaction. Les informations provenant de ces sous-détecteurs sont combinées pour déterminer le type, l'énergie et la direction des particules de l'état final de la collision, pour pouvoir mesurer les propriétés de celle-ci, et par exemple determiner si une particule instable encore inconnue a été produite.

Nous allons nous intéresser plus spécifiquement au détecteur CMS [3]. Celui-ci dispose :

— d'un champ magnétique, pour courber la trajectoire des particules chargées ;

— d'un trajectographe, pour reconstruire la trajectoire des particules chargées, et ainsi obtenir la charge et l'impulsion ;

— d'un calorimètre électromagnétique (ECAL) [4], constitué d'un cristal de tungstate de plomb, permettant de collecter les dépôts d'énergie des particules, principalement électrons et photons, mais aussi hadrons chargés et neutres) ;

— d'un calorimètre hadronique (HCAL) [5], composé de plusieurs couches d'absorbeur en laiton et de carreaux scintillateurs en plastique, avec une segmentation grossière. La resolution du HCAL pour la mesure de l'energie E d'un hadron est de l'ordre de $100\%\sqrt{(E/\mathrm{GeV})}$;

— de chambres à muons, qui permettent l'dentification de ces particules, les seules à pouvoir y parvenir.

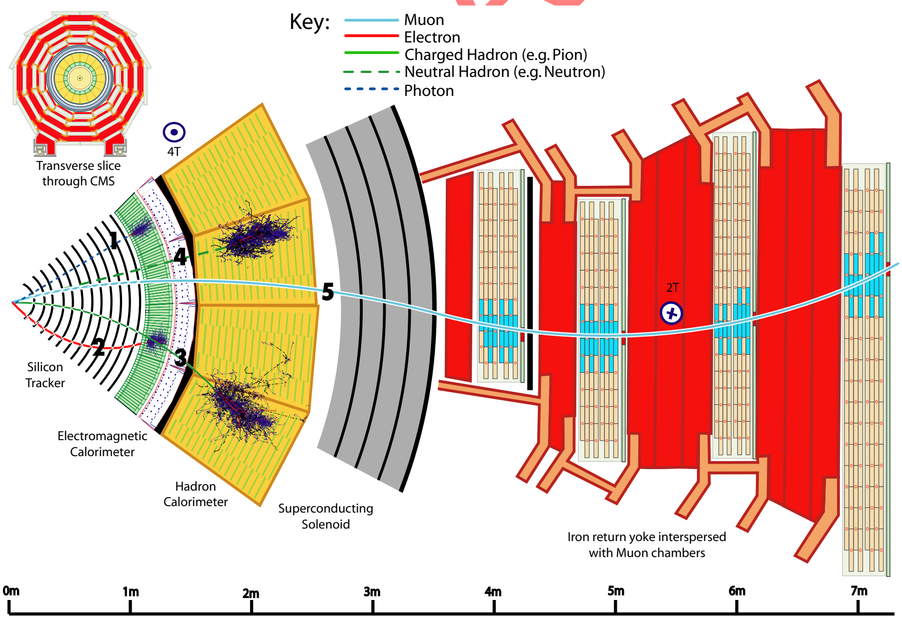


Figure 1 – Une esquisse des interactions spécifiques des particules dans une tranche transversale du détecteur CMS.

Détaillons alors le comportement des particules :

— photons (exemple 1 dans la Fig. 1) :

  — déposent leur énergie dans ECAL ;

— e^+, e^- (exemple 2 dans la Fig. 1) :

  — produisent une trace dans le trajectographe ;

  - déposent leur énergie dans ECAL ;
- hadrons chargés (exemple 3 dans la Fig. 1) :
  - produisent une trace dans le trajectographe ;
  - déposent en minorité des cas leur énergie dans ECAL ;
  - déposent leur énergie dans HCAL ;
  - fin de course dans HCAL ;
- hadrons neutres (exemple 4 dans la Fig. 1) :
  - déposent leur énergie dans le HCAL ;
  - déposent en minorité des cas leur énergie dans ECAL ;
  - fin de course dans HCAL ;
- μ^+, μ^- (exemple 5 dans la Fig. 1) :
  - produisent une trace dans le trajectographe ;
  - traversent ECAL, HCAL ;
  - chambre à muons.

À noter que dans notre étude seuls les hadrons neutres nous intéressent.

La connaissance des dépôts d'énergies et du comportement des particules dans les différentes parties du détecteur nous permette de reconnaître et distinguer les particules, cette opération s'appelle le *Particle Flow (PF)*. Cependant, il est aussi nécessaire d'estimer l'énergie des particules (E_{true}) à l'aide d'une calibration des calorimètres. En effet, ces derniers ne présentent pas une réponse linéaire et la somme des énergies dans les calorimètres ne correspond pas à l'énergie de la particule. Cette énergie de calibration sera notée E_{calib}.

En première approximation, nous déterminerons l'énergie calibrée par une fonction linéaire de l'énergie lue dans le ECAL (énergie notée par la suite E_{ecal}) et de celle lue dans le HCAL (notée par la suite H_{ecal}). Cette méthode sera présentée dans la section 3.

Ensuite, ce rapport présente de nouvelles techniques de calibration qui permettent de prendre en compte la non-linearité des calorimètres. Ces techniques seront présentées dans les sections 4 et 5. Enfin, dans la section 6 nous comparerons ces méthodes.

2 Production de l'échantillon

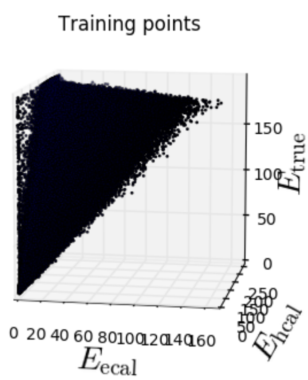


Figure 2 – Le nuage de points $(E_{\text{ecal}}, E_{\text{hcal}}, E_{\text{true}})$

- paragraphe de Colin sur la simulation

Les données simulées sont stockées dans un fichier *.root*. Pour des soucis de compatibilité, j'ai voulu écarter *Root*, qui est à la fois un programme et une librairie C++ pour tout faire en Python. J'ai donc écrit un programme [6] pour extraire les données du fichier initial pour les déplacer dans un nouveau dans un fichier binaire importable facilement par Python et qui contient pour chaque hadron simulée : son énergie réelle (E_{true}), son impulsion, l'énergie déposée dans ECAL (E_{ecal}), l'énergie déposée dans HCAL (E_{hcal}), la pseudo-rapidité.

On séparera et traitera différemment les événements qui ont $E_{\text{ecal}} = 0$. Ces événements sont liés à des particules qui ont interagit avec le détecteur hardronique mais pas avec le détecteur électromagnétique (cf Fig.1). Cette séparation se justifie par le fait que modéliser les dépôts d'énergie dans les deux calorimètres pour en conclure ce qui se passe dans le cas particulier où il n'y a des dépôts que dans un amène un biais. Ainsi, à chaque construction de calibration, on créera en fait deux modèles.

De plus, la simulation crée un palier car les particules sont limitées à $E_{\text{true}} = 200\text{GeV}$, si l'on regarde par exemple sur la Fig. 3 ce qui se passe dans le plan $E_{\text{ecal}} = 0$, on voit bien ce palier qui apparaît vers 150GeV, il est donc ici impossible de déduire une énergie de calibration à partir des points au-delà de cette limite . Avec ce jeu de donnée, nous allons fixer une limite $E_{\text{ecal}} + E_{\text{hcal}} =$ (lim $= 150\text{GeV}$).

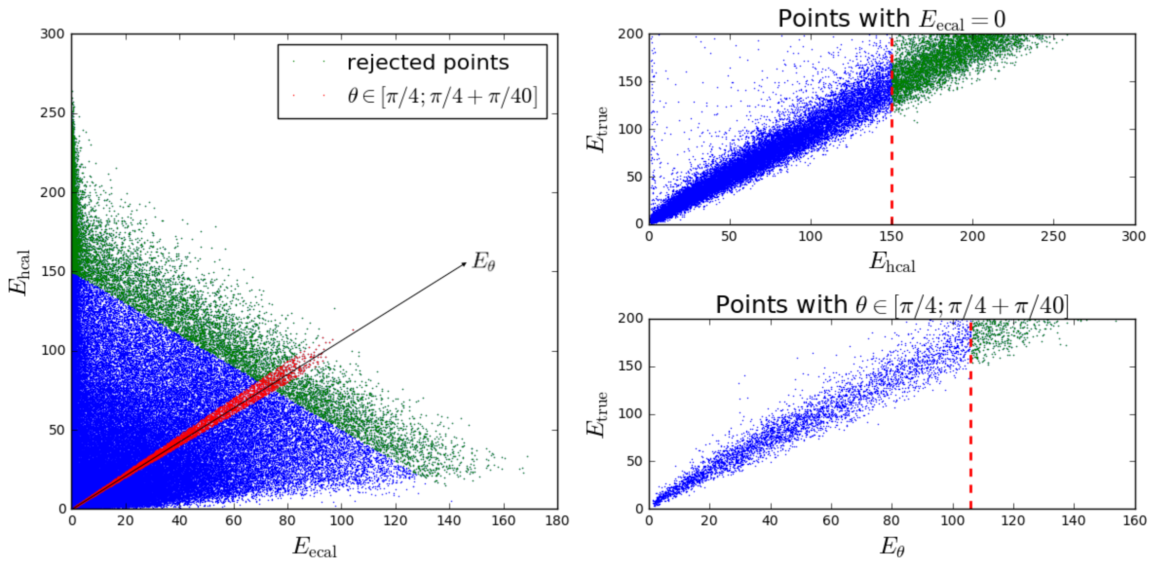


FIGURE 3 – On place une limite à $E_{\text{ecal}} + E_{\text{hcal}} = 150$. Les points verts sont les points rejetés.

3 Calibration par régression linéaire

Comme première calibration, j'ai utilisé une méthode simple, la régression linéaire. Il s'agit de supposer qu'il existe une relation

$$E_{\text{true}} = a_1 E_{\text{ecal}} + a_2 E_{\text{hcal}} + b \tag{1}$$

et de trouver les coefficient a_1, a_2, b optimaux pour que le modèle coïncide au mieux aux données d'entrainement. Au niveau de la programmation, j'ai utilisé la librairie de *Sciki Learn* [7] qui se base une méthode des moindres carrés, c'est à dire dans notre cas, trouver les coefficients qui, pour un ensemble de données d'entraînement ($E^n_{\text{ecal}}, E^n_{\text{hcal}}, E^n_{\text{true}}$) minimiseront :

$$\epsilon = \sum_n |E_{\text{true}} - a_1 E_{\text{ecal}} - a_2 E_{\text{hcal}} - b|^2 \tag{2}$$

Comme j'ai constaté qu'à faible $E_{\text{ecal}}, E_{\text{hcal}}$ il y a quelques points à très haut E_{true} qui correspondent à des défauts de la détection des énergies dans les calorimètres, je les ai enlevés des données d'entraînement pour la regression linéaire (cf Fig. 4).

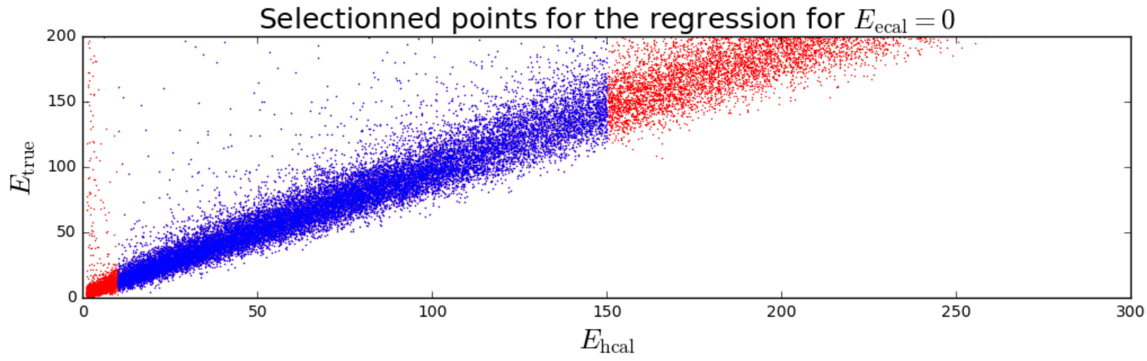
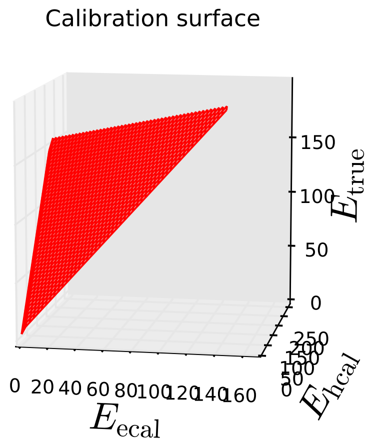


Figure 4 – En bleu, les points sélectionnés pour faire la régression linéaire.

Nous pouvons à présent effectuer la regression linéaire sur un premier jeu de donné d'entrainement [8] en ne prenant en compte dans la régression que les points tels que $\lim_{\text{min}} < E_{\text{ecal}} + E_{\text{hcal}} < \lim_{\text{max}}$, avec $\lim_{\text{min}} = 10, \lim_{\text{min}} = 150$. Nous obtenons alors :

$$\begin{cases} E_{\text{calib}} = 0.95359395 E_{\text{hcal}} + 5.9057809 & E_{\text{ecal}} = 0 \\ E_{\text{calib}} = 0.9876819 E_{\text{hcal}} + 1.3081561 E_{\text{ecal}} + 8.5016036 & E_{\text{ecal}} \neq 0 \end{cases}$$



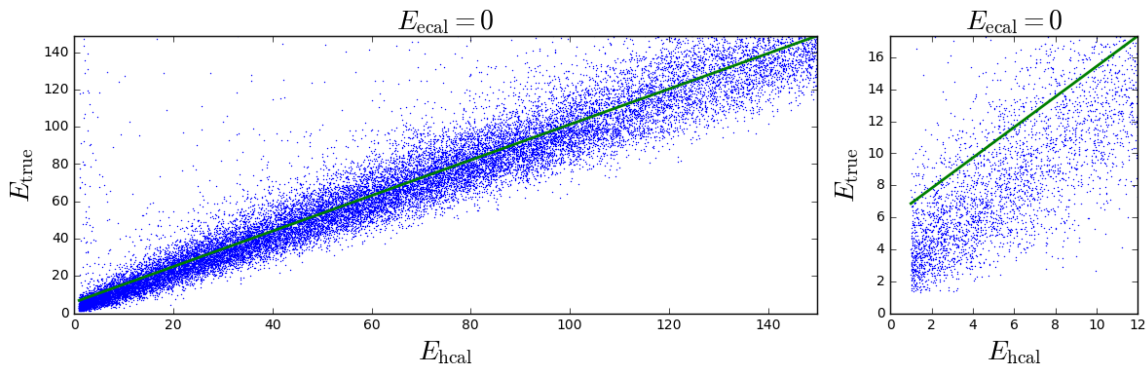


Figure 5 – Courbe de calibration pour $E_{\text{ecal}} = 0$.

Nous constatons en particulier dans le cas $E_{\text{ecal}} = 0$ que la courbe ne passe pas par le le coeur du nuage de point à faible $E_{\text{hcal}} = 0$. Maintenant que la régression est faite, nous allons calibrer un second jeu de données et afficher $E_{\text{calib}}/\text{E}_{\text{true}}$ pour ce nouveau jeu de donnée. $E_{\text{calib}}/\text{E}_{\text{true}}$ doit être le plus proche possible de 1.

Sur la Fig. 6, à gauche, nous constatons que comme prévu la régression linéaire est mauvaise à faible E_{hcal} car en moyenne, $E_{\text{calib}}/E_{\text{true}}$ n'est pas proche de 1. Plus intéressant, la figure de droite met en avant les non-linéarités du nuage de point.

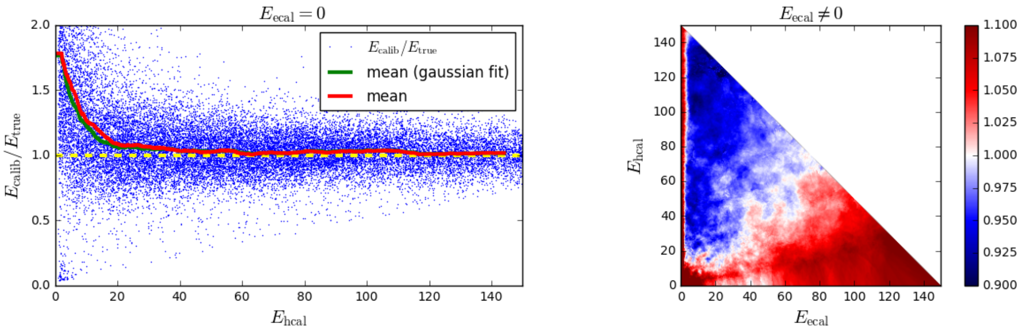


FIGURE 6 – $E_{\text{calib}}/E_{\text{true}}$ en fonction de E_{ecal} et E_{hcal}.

Sur la Fig. 7 nous constatons également que la calibration s'écarte $E_{\text{calib}}/E_{\text{true}}$ est trop écartée de 1.

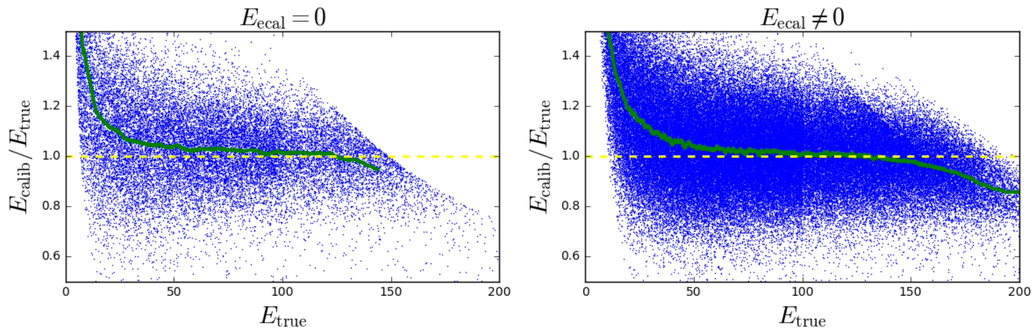


FIGURE 7 – $E_{\text{calib}}/E_{\text{true}}$ en fonction de E_{true}.

Il faut donc proposer une méthode qui prend en compte la non-linéarité entre les 3 variables.

4 Méthode non paramétrique binnée

Comme nous l'avons vu précédemment, il faut une calibration qui prenne en compte les non-linéarités. Ici, l'idée est de découper le plan $(E_{\text{ecal}}, E_{\text{hcal}})$ en carré et de calculer la moyenne des E_{true} dans chaque carré qui sera la valeur E_{calib}. Ainsi pour prédire une énergie de E^i_{calib} pour un point $(E^i_{\text{ecal}}, E^i_{\text{hcal}})$, nous allons regarder dans quel carré il se trouve et retourner la valeur d'énergie calibrée correspondante, faisant apparaître ainsi des "legos".

- méthode binné
- illustration :

->Résultat de la calibration - lancer la calibration ref Git [10]
- paramètres
-> plot surface, courbe de calibration
- on calibre un second jeu de données
-> plot ecalib/etrue
- biais lié aux bins

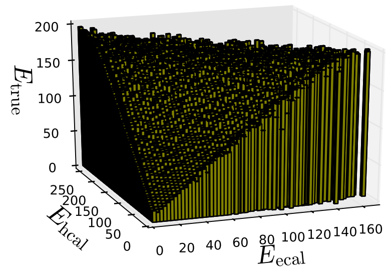


Figure 8 – Le nuage de points modélisé par des legos

5 Méthodes basées sur les plus proches voisins

5.1 Moyenne pondérée

-Principe général de l'algorithme
- trouver les plus proches voisin : sckitlearn [11], algorithmes de recherches optimisés

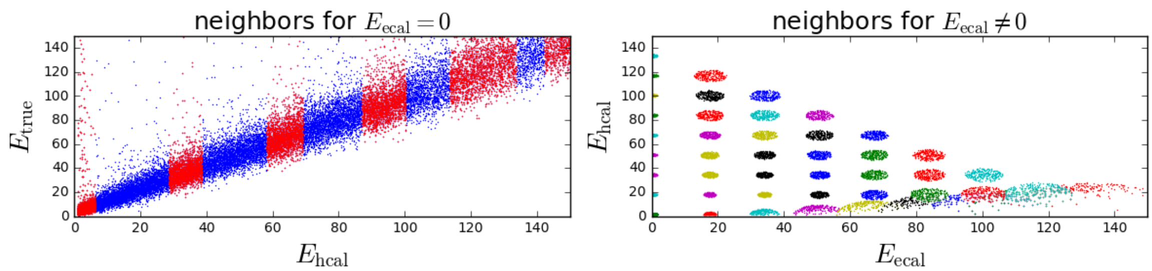


Figure 9 – $n_{voisins} = 2000$ pour $E_{\mathrm{ecal}} = 0$, $n_{voisins} = 250$ pour $E_{\mathrm{ecal}} \neq 0$

- moyenne pondérée par une gaussienne
- pas assez lissé, des particules à fort Etrue mal détectées perturbent la calibration, il faut les enlever
-Résultat de la calibration
- créer la calibration, doc git [12] - paramètres :
$\mathrm{lim} = 150, \mathrm{n}_{\mathrm{neighbors,Ecal=0}} = 2000, \mathrm{n}_{\mathrm{neighbors,Ecal\neq 0}} = 250, \mathrm{lim} = 150, \sigma = 5$
weights = 'gaussian'
-> plot surface, courbe de calibration
- on calibre un second jeu de données
-> plot ecalib/etrue

5.2 Nettoyage gaussien

5.2.1 Principe général de l'algorithme

- on enlève les points éloignés du coeur de la distribution
- principe de l'algo
- comment fait-on un fit
-scipy [9]
expliquer :

- barre d'erreur
- minimisation du chi2
- un bon chi2 réduit ?
- quand nous ferons un fit gaussien ce sera toujours le même principe
- interpolation [13]

5.2.2 Résultat de la calibration

- doc git [14] - paramètres :
$\mathrm{lim} = 150, \mathrm{n}_{\mathrm{neighbors,Ecal=0}} = 2000, \mathrm{n}_{\mathrm{neighbors,Ecal\neq 0}} = 250, \mathrm{lim} = 150, \sigma = 5$
weights = 'gaussian'
algorithm = 'auto'
energystep = 1
kind = 'cubic'
cut = 2
-> plot surface, courbe de calibration
- on calibre un second jeu de données
-> plot ecalib/etrue
- mise en évidence des non linéarités

5.3 Fit gaussien

5.3.1 Principe général de l'algorithme

- principe de l'algo
- Ecalib = μ

5.3.2 Résultat de la calibration

- doc git [15] - paramètres :
$\mathrm{lim} = 150, \mathrm{n}_{\mathrm{neighbors,Ecal=0}} = 2000, \mathrm{n}_{\mathrm{neighbors,Ecal\neq 0}} = 250, \mathrm{lim} = 150, \sigma = 5$
algorithm = 'auto'
kind = 'cubic'
energystep$_e cal_e q_0 = 1$
$energystep_e cal_n eq_0 = 5$
$- > plotsurface, courbedecalibration$
$- oncalibreunsecondjeudedonnes$
$- > plotecalib/etrue$
$- miseenvidencedesnonlinarits$

6 Comparaison des méthodes

6.1 Méthodes des plus proches voisins

7 Annexes

7.1 Fonctions utiles du programme

Références

[1] ATLAS Collaboration. Observation of a new particle in the search for the standard model higgs boson with the ATLAS detector at the LHC. *Physics Letters B*, 716(1) :1 – 29, 2012.

[2] CMS Collaboration. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Physics Letters B*, 716(1) :30 – 61, 2012.

[3] CMS Collaboration. The CMS experiment at the cern LHC. *Journal of Instrumentation*, 3(08) :S08004, 2008. http://stacks.iop.org/1748-0221/3/i=08/a=S08004.

[4] CMS Collaboration. The CMS electromagnetic calorimeter project : technical design report. *Technical Design Report CMS*, 1997. https://cds.cern.ch/record/349375.

[5] CMS Collaboration. CMS : The hadron calorimeter technical design report. *Technical Design Report CMS*, 1997. https://cds.cern.ch/record/349375.

[6] S. Niang. *Root file to python*. https://sniang.github.io/particle_flow_calibration.

[7] Scikit Learn. *Generalized Linear Models*. scikit-learn.org.

[8] S. Niang. *Classe LinearRegression*. https://github.com/sniang/particle_flow_calibration.

[9] Scipy. *Optimization*. docs.scipy.org.

[10] S. Niang. *Classe CalibrationLego*. github.com/sniang/particle_flow_calibration.

[11] Scikit Learn. *Nearest Neighbors*. scikit-learn.org.

[12] S. Niang. *Classe KNN*. github.com/sniang/particle_flow_calibration.

[13] Scipy. *Interpolation*. docs.scipy.org.

[14] S. Niang. *Classe KNNGC*. github.com/sniang/particle_flow_calibration.

[15] S. Niang. *Classe KNNGF*. github.com/sniang/particle_flow_calibration.