

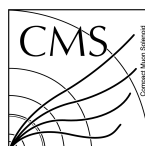
---

# 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*

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Production de l'échantillon</b>	<b>3</b>
<b>3</b>	<b>Calibration par régression linéaire</b>	<b>4</b>
<b>4</b>	<b>Méthode non paramétrique binnée</b>	<b>6</b>
<b>5</b>	<b>Méthodes basées sur les plus proches voisins</b>	<b>8</b>
5.1	Moyenne pondérée . . . . .	8
5.2	Nettoyage gaussien . . . . .	10
5.3	Fit gaussien . . . . .	12
<b>6</b>	<b>Comparaison des méthodes</b>	<b>14</b>
6.1	Méthodes des plus proches voisins . . . . .	14
<b>7</b>	<b>Annexes</b>	<b>14</b>
7.1	Fonctions utiles du programme . . . . .	14

DRAFT

# 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 déterminer 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 résolution du HCAL pour la mesure de l'énergie  $E$  d'un hadron est de l'ordre de  $100\% \sqrt{(E/\text{GeV})}$  ;
- de chambres à muons, qui permettent l'identification de ces particules, les seules à pouvoir y parvenir.

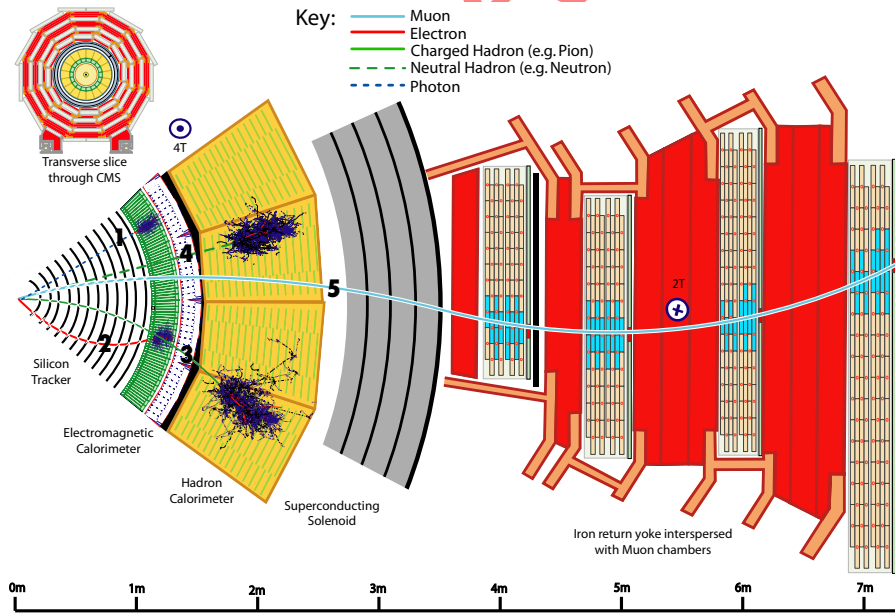


FIGURE 1 – Une esquisse des interactions spécifiques des particules dans une tranche transversale du détecteur CMS.

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 ;
  - finissent leur course dans HCAL ;
- hadrons neutres (exemple 4 dans la Fig. 1) :
  - déposent leur énergie dans le HCAL ;
  - déposent dans la minorité des cas leur énergie dans ECAL ;
  - finissent leur course dans HCAL ;
- $\mu^+, \mu^-$  (exemple 5 dans la Fig. 1) :
  - produisent une trace dans le trajectographe ;
  - traversent ECAL, HCAL ;
  - sont visibles dans la chambre à muons.

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

La connaissance des dépôts d'énergie et du comportement des particules dans les différentes parties du détecteur nous permettent 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 ( $E_{\text{true}}$ ) 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  $E_{\text{calib}}$ .

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

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

## 2 Production de l'échantillon

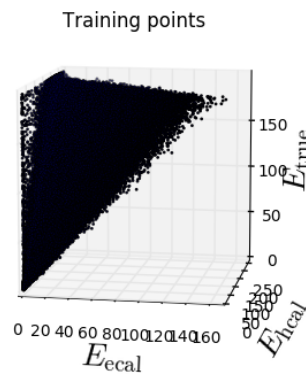


FIGURE 2 – Le nuage de points  $(E_{\text{ecal}}, E_{\text{hcal}}, E_{\text{true}})$

- paragraphe de Colin sur la simulation

Les données simulées sont stockées dans un fichier *.root*. Par souci 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 fichier binaire importable facilement par Python et qui contient, pour chaque hadron simulé : son énergie réelle ( $E_{\text{true}}$ ), son impulsion, l'énergie déposée dans ECAL ( $E_{\text{ecal}}$ ), l'énergie déposée dans HCAL ( $E_{\text{hcal}}$ ), sa pseudo-rapidité.

On séparera et traitera différemment les événements qui ont  $E_{\text{ecal}} = 0$ . Ces événements sont liés à des particules qui ont interagi avec le détecteur hadronique mais pas avec le détecteur électromagnétique (cf Fig.1). Cette séparation se justifie, car 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 de dépôt que dans l'un d'eux 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 données, nous allons fixer une limite  $E_{\text{ecal}} + E_{\text{hcal}} = (\text{lim} = 150\text{GeV})$ .

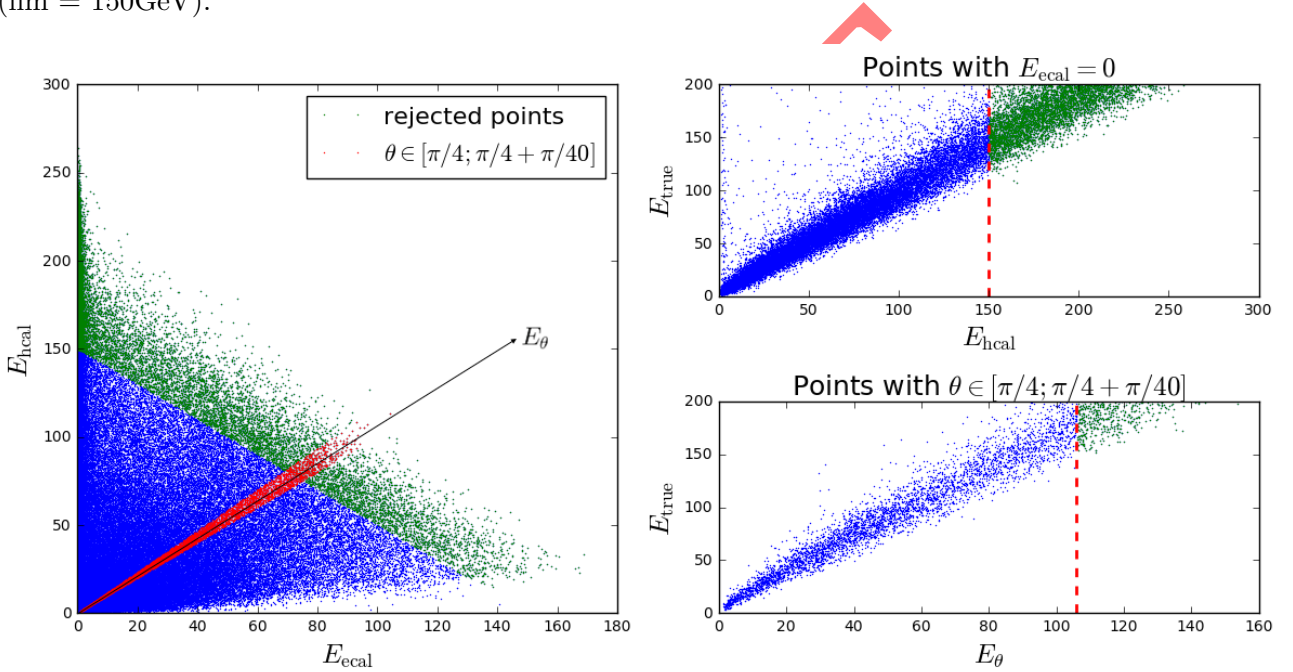


FIGURE 3 – On place une limite à  $E_{\text{ecal}} + E_{\text{hcal}} = 150$ . Les points verts sont les points rejetés.

### 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

$$E_{\text{true}} = a_1 E_{\text{ecal}} + a_2 E_{\text{hcal}} + b \quad (1)$$

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

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

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

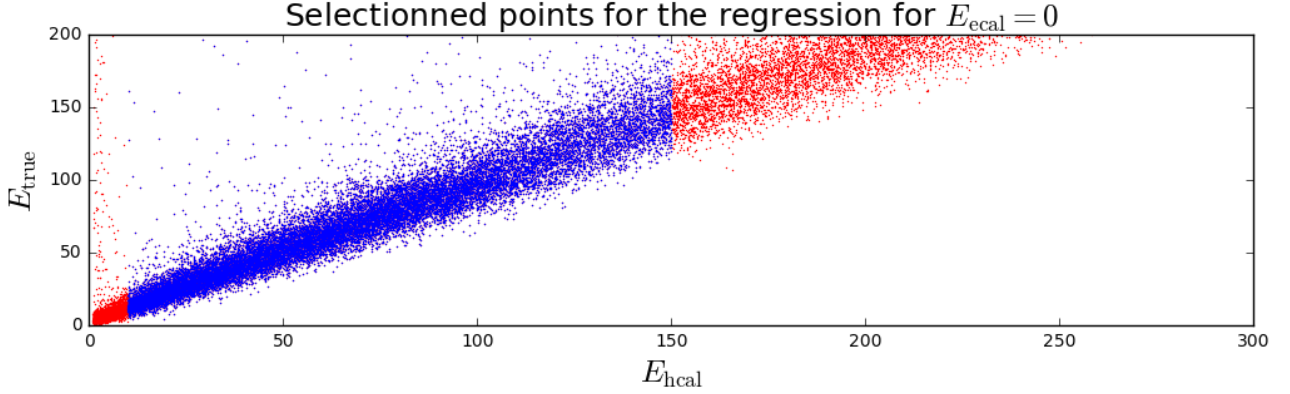


FIGURE 4 – En bleu, les points sélectionnés pour faire la régression linéaire.

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

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

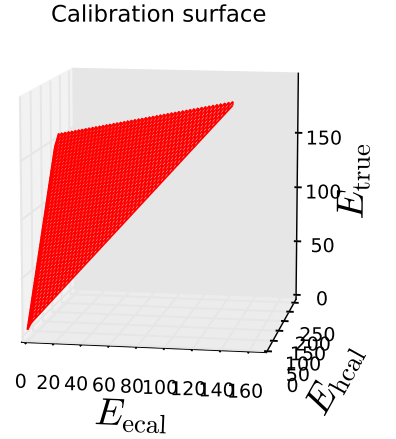


FIGURE 5 – Nuage de points modélisé par un plan

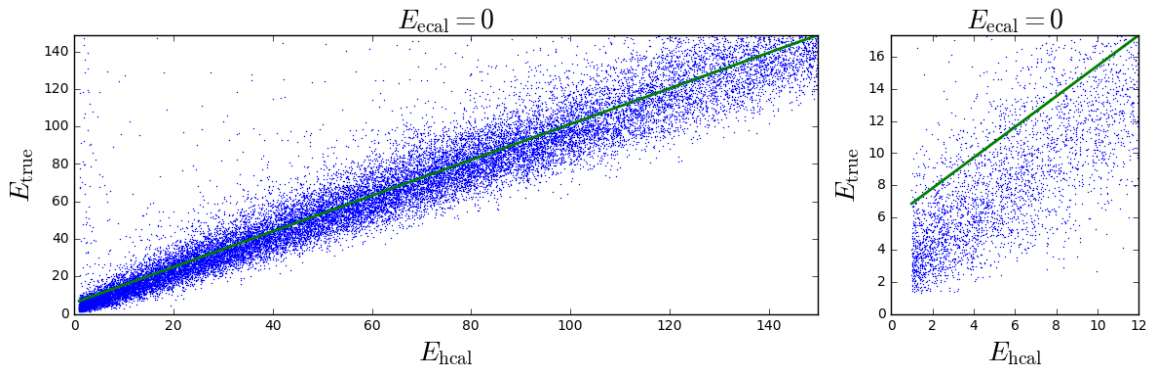


FIGURE 6 – Courbe de calibration pour  $E_{\text{ecal}} = 0$ .

Nous constatons en particulier que dans le cas  $E_{\text{ecal}} = 0$ , la courbe ne passe pas par le cœur 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  $E_{\text{calib}}/E_{\text{true}}$  pour ce nouveau jeu de données.  $E_{\text{calib}}/E_{\text{true}}$  doit être le plus proche possible de 1.

Sur la Fig. 7, à gauche, nous constatons comme prévu que la régression linéaire est mauvaise à faible  $E_{\text{hcal}}$  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.

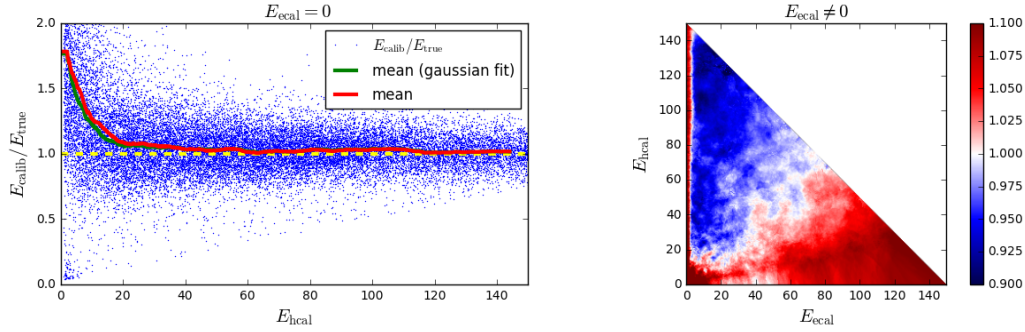


FIGURE 7 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{ecal}}$  et  $E_{\text{hcal}}$ .

Sur la Fig. 8 nous constatons également que la calibration dévie :  $E_{\text{calib}}/E_{\text{true}}$  est trop écarté de 1.

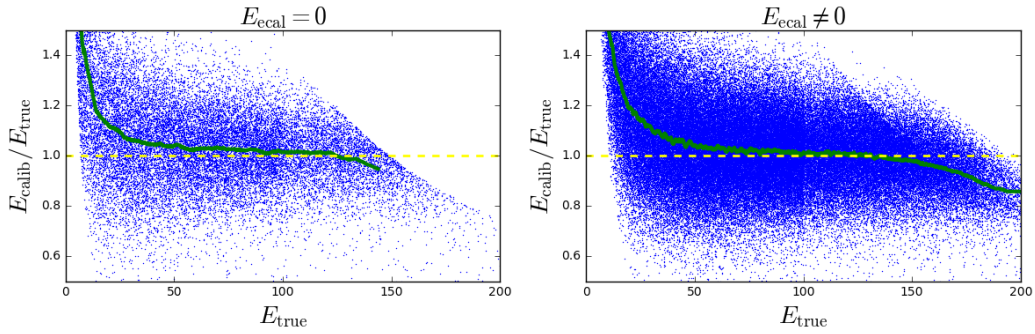


FIGURE 8 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{true}}$ .

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

## 4 Méthode non paramétrique binnée

Ici, l'idée est de découper le plan  $(E_{\text{ecal}}, E_{\text{hcal}})$  en carrés et de calculer la moyenne des  $E_{\text{true}}$  dans chaque carré qui sera la valeur  $E_{\text{calib}}$ . Ainsi pour prédire une énergie de  $E_{\text{calib}}$  pour un point  $(E_{\text{ecal}}^i, E_{\text{hcal}}^i)$ , nous allons regarder dans quel carré il se trouve et retourner la valeur d'énergie calibrée correspondante. Nous pouvons voir sur la Fig. 10 une illustration où la hauteur de chaque brique correspond à l'énergie calibrée d'un carré.

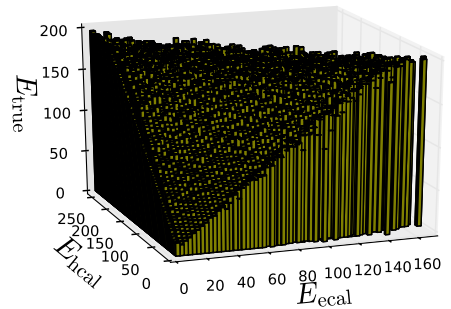


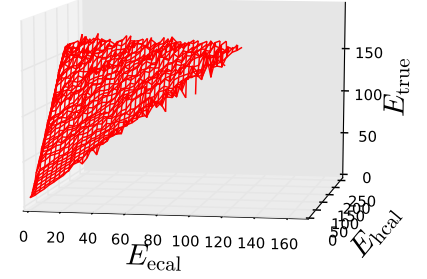
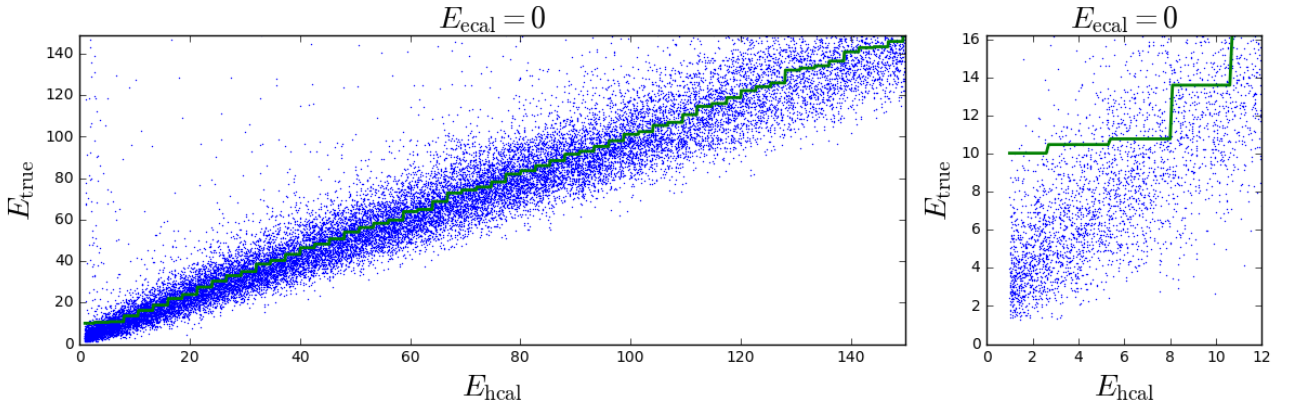
FIGURE 9 – Illustration de la calibration avec  $100 \times 100$  carrés.



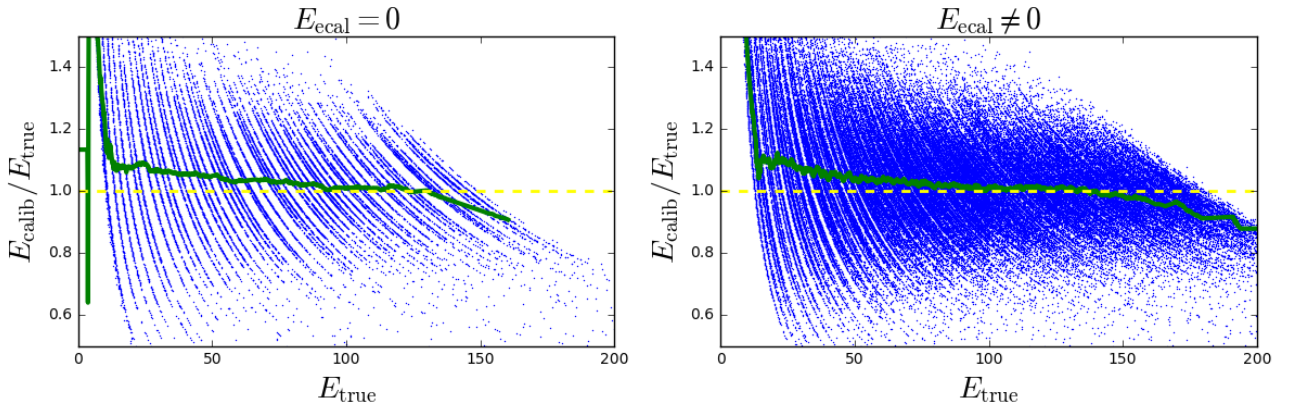
L'avantage de cette calibration est de prendre en compte ce qui se passe localement dans la distribution contrairement à la régression linéaire, mais elle a pour désavantage d'être liée au pas des carrés. Un pas trop grand fait perdre en précision, un pas trop petit laisse des trous dans la position des briques (cf Fig. 10), et fait exploser le temps de calcul.

Construisons la calibration [10] à l'aide d'un premier jeu de données, avec  $100 \times 100$  carrés. Premièrement, nous constatons que la surface est "cabossée", et si nous regardons ce qui se passe dans le plan  $E_{\text{ecal}} = 0$  (Fig. 11), nous voyons que cette méthode binnée donne des énergies calibrées  $E_{\text{calib}}$  égales pour des ensembles de points, par paliers. Deuxièmement, nous constatons une fois de plus que la courbe de calibration ne passe pas par le coeur de la distribution à bas  $E_{\text{hcal}}$  :  $E_{\text{calib}}$  y est sur-évaluée.

Calibration surface

FIGURE 10 – Surface de calibration avec  $100 \times 100$  carrés.FIGURE 11 – Courbe de calibration pour  $E_{\text{ecal}} = 0$ .

Effectuons la calibration d'un second jeu de données :

FIGURE 12 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{true}}$ . Nous voyons clairement l'apparition d'une structure, liée au caractère binné de la méthode.

Nous remarquons que la présence de paliers dans la courbe de calibration biaise la calibration en



faisant apparaître une structure (des hyperboles) dans  $E_{\text{calib}}/E_{\text{true}}$ , comme nous pouvons le voir dans la Fig. 12. Cette structure est liée aux points possédant la m<sup>a</sup>me énergie de calibration, contrairement au cas de la régression linéaire (cf Fig. 8). Nous constatons également sur la Fig. 12 que  $E_{\text{calib}}/E_{\text{true}}$  est en moyenne constamment  $> 1$ , ce qui montre qu'il y a une tendance à la sur-évaluation de  $E_{\text{calib}}$ .

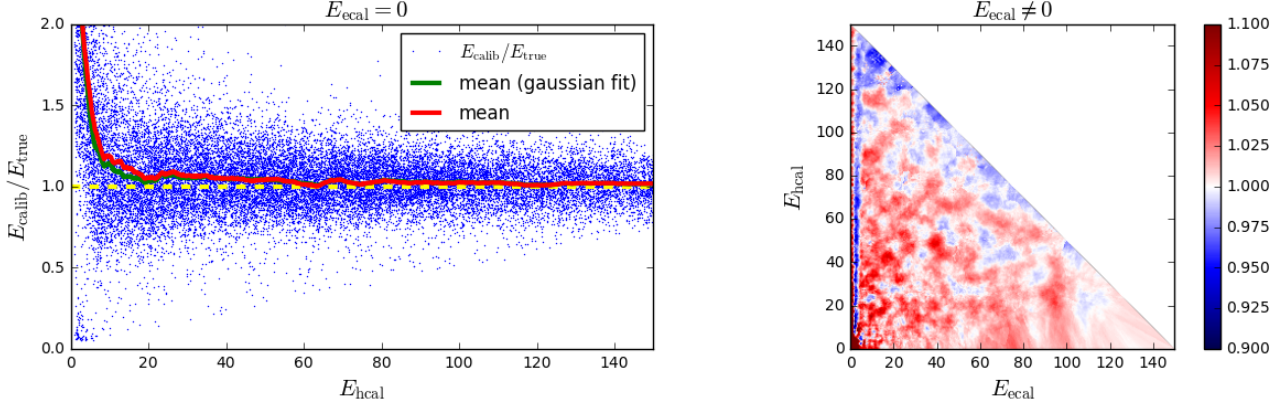


FIGURE 13 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{eal}}$  et  $E_{\text{hcal}}$ .

La Fig. 13 nous montre une fois de plus que  $E_{\text{calib}}/E_{\text{true}}$  est globalement sur-évaluée, mais que cette fois-ci, nous avons pris en compte les non-linéarités. Il nous faut alors une méthode qui reprenne l'idée de moyenner des  $E_{\text{true}}$  en enlevant les contraintes et biais provenant des bins.

## 5 Méthodes basées sur les plus proches voisins

### 5.1 Moyenne pondérée

Nous utilisons encore des données simulées pour effectuer la calibration : chaque particule simulée  $i$  est vue comme un point d'un espace tridimensionnel possédant des coordonnées  $(E_{\text{eal}}^i, E_{\text{hcal}}^i, E_{\text{true}}^i)$ . Pour trouver l'énergie calibrée d'un point de coordonnées  $(E_{\text{eal}}^0, E_{\text{hcal}}^0)$  :

- on recherche ses  $k$  plus proches voisins dans le plan  $(E_{\text{eal}}, E_{\text{hcal}}) \rightarrow (E_{\text{eal}}^i, E_{\text{hcal}}^i), i \in [1, \dots, k]$
- on effectue une moyenne pondérée des  $E_{\text{true}}^i$  de ces plus proches voisins  $\rightarrow E_{\text{calib}}^0$  : l'énergie calibrée.

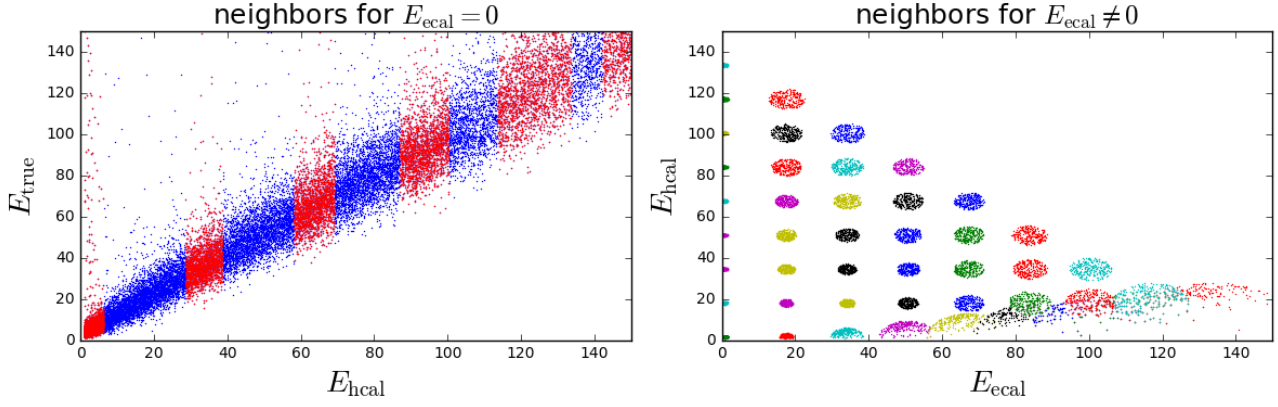
La moyenne pondérée va donc s'exprimer ainsi :

$$E_{\text{calib}}^0 = \frac{\sum_{i=1}^k g(E_{\text{eal}}^i, E_{\text{hcal}}^i) \times E_{\text{true}}^i}{\sum_{i=1}^k g(E_{\text{eal}}^i, E_{\text{hcal}}^i)} \quad (3)$$

Dans notre cas nous avons pris pour facteur de pondération  $g$  la distribution gaussienne  $g(\vec{x}) = \exp -\frac{1}{2}(\frac{(\vec{x}-\vec{x}^0)^2}{\sigma^2})$ , pour donner plus d'importance aux plus proches des  $k$  plus proches voisins.

Pour trouver les plus proches voisins, j'ai utilisé la librairie *Scikit Learn* [11], l'avantage de cette librairie, est qu'elle propose différents algorithmes optimisés de recherche. Lors de la construction de la calibration [12], j'ai d'ailleurs laissé le choix de l'algorithme de recherche.

Si l'on regarde la Fig. 14, on voit que l'on sélectionne les points à moyenner dans des cercles dont le rayon varie en fonction de la densité de points. Il y a un problème à bas  $E_{\text{hcal}}$  et haut  $E_{\text{eal}}$  car nous manquons de points.

FIGURE 14 –  $n_{voisins} = 2000$  pour  $E_{ecal} = 0$ ,  $n_{voisins} = 250$  pour  $E_{ecal} \neq 0$ 

Construisons alors la calibration [12] à l'aide des paramètres suivants :

$$\begin{cases} \text{lim} & = 150 \\ n_{\text{neighbors}, E_{ecal}=0} & = 2000 \\ n_{\text{neighbors}, E_{ecal} \neq 0} & = 250 \\ \text{lim} & = 150 \\ \sigma & = 5 \\ \text{weights} & = \text{'gaussian'} \\ \text{algorithm} & = \text{'auto'} \end{cases}$$

Calibration surface

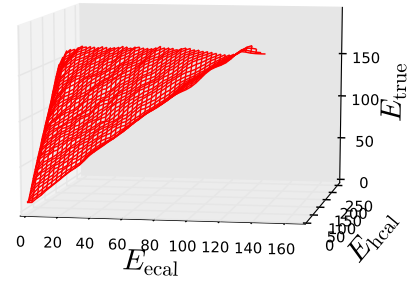
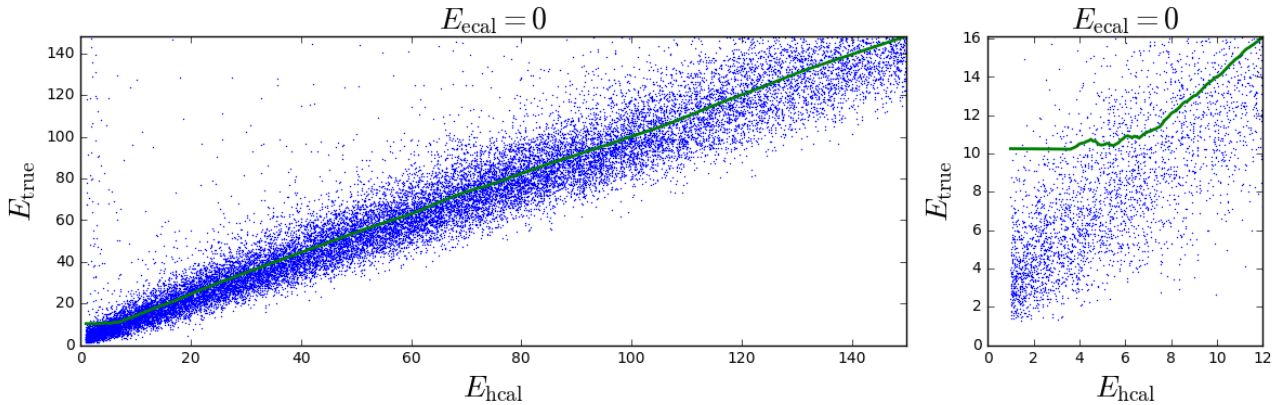


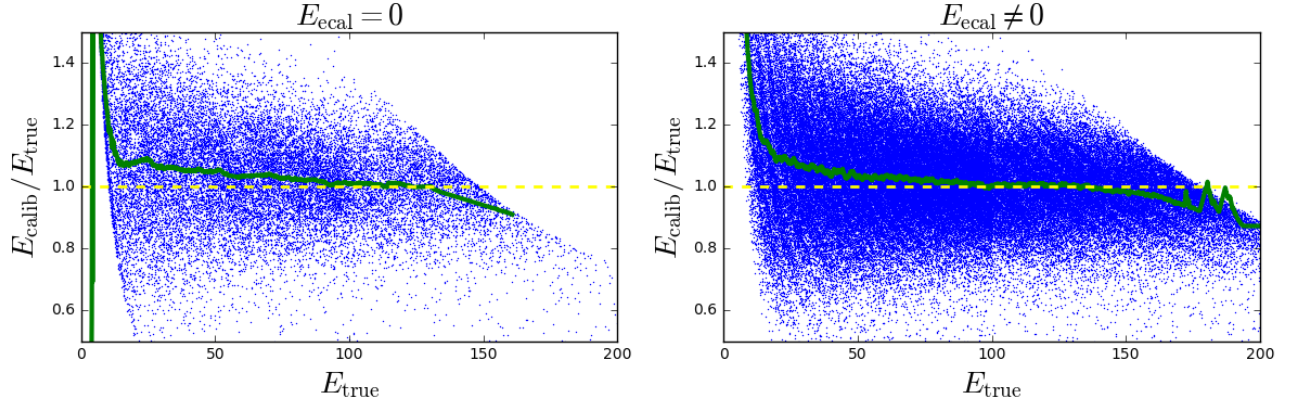
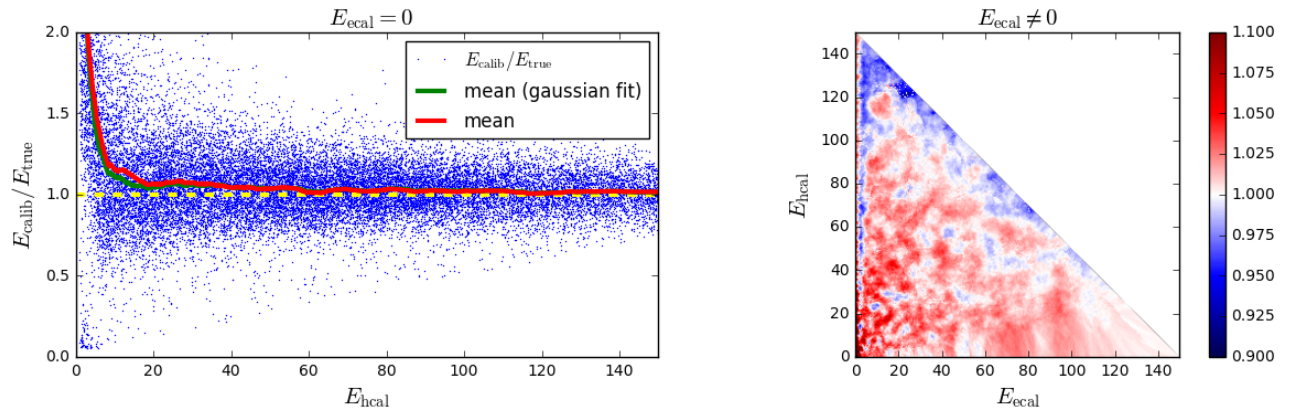
FIGURE 15 – Surface de calibration.

FIGURE 16 – Courbe de calibration pour  $E_{ecal} = 0$ .

Nous constatons ici que la surface est beaucoup plus lisse qu'avec la méthode précédente, mais en regardant le cas particulier de  $E_{ecal} = 0$ , nous constatons qu'une fois encore, à bas  $E_{hcal}$ , la courbe de calibration ne passe pas par le coeur de la gerbe. Cela est dû à des points aberrants qui correspondent à des particules de haute énergie qui ne déposent pas ou peu leur énergie dans les calorimètres. Il faudra trouver un moyen de les écarter par la suite.

Effectuons alors la calibration d'un second jeu de données :

Nous constatons que l'énergie de calibration est toujours sur-estimée, et particulièrement à bas  $E_{ecal}$  et  $E_{hcal}$  ( cf. 18).

FIGURE 17 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{true}}$ .FIGURE 18 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{ecal}}$  et  $E_{\text{hcal}}$ .

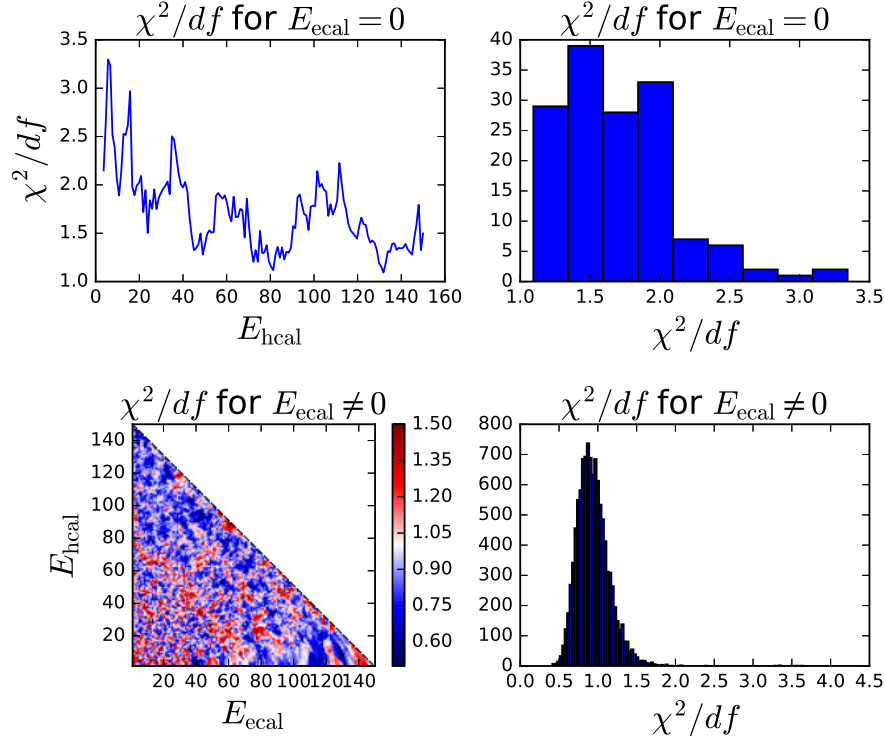
## 5.2 Nettoyage gaussien

Cette méthode est assez similaire à la précédente. Elle se base sur la constatation que la distribution en énergie vraie des paquets de plus proches voisins est une distribution gaussienne. En utilisant la méthode des moindres carrés, nous allons donc trouver les paramètres de la gaussienne en question et ne prendre en compte les plus proches voisins dont l'énergie vraie est  $\mu - c\sigma \leq E_{\text{true}}^i \leq \mu + c\sigma$  (nous prenons par défaut  $c = 2$ ), avec  $\mu, \sigma$  la moyenne et l'écart type de la distribution gaussienne.

Principe de l'algorithme :

- on considère des points  $(E_{\text{ecal}}^{0,j}, E_{\text{hcal}}^{0,j})$  où nous allons évaluer l'énergie calibrée.
- pour chaque  $(E_{\text{ecal}}^{0,j}, E_{\text{hcal}}^{0,j})$  :
  - on recherche ses  $k$  plus proches voisins dans le plan  $(E_{\text{ecal}}, E_{\text{hcal}}) \rightarrow (E_{\text{ecal}}^i, E_{\text{hcal}}^i), i \in [1, \dots, k]$
  - on trouve la gaussienne correspondante  $\mu - c\sigma \leq E_{\text{true}}^i \leq \mu + c\sigma$
  - on ne conserve que les voisins dont :  $\mu - c\sigma \leq E_{\text{true}}^i \leq \mu + c\sigma$
  - on effectue une moyenne pondérée de l'énergie vraie de ces plus proches voisins  $\rightarrow E_{\text{calib}}^0$  : l'énergie calibrée
- on effectue une interpolation pour donner une valeur d'énergie calibrée quelque soit  $(E_{\text{ecal}}^0, E_{\text{hcal}}^0)$ 
  - on enlève les points éloignés du coeur de la distribution
- principe de l'algo

- comment fait-on un fit
- scipy [9]
- Efficacité du fit : expliquer :

FIGURE 19 – Le  $\chi^2$  réduit pour chaque fit effectué.

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

Construisons alors la calibration [14] à l'aide des paramètres suivants :

lim	= 150
$n_{\text{neighbors}, E_{ecal}=0}$	= 2000
$n_{\text{neighbors}, E_{ecal} \neq 0}$	= 250
lim	= 150
$\sigma$	= 5
weights	= 'gaussian'
algorithm	= 'auto'
energystep	= 1
kind	= 'cubic'
cut	= 2

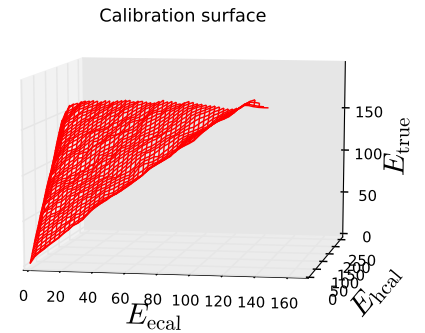
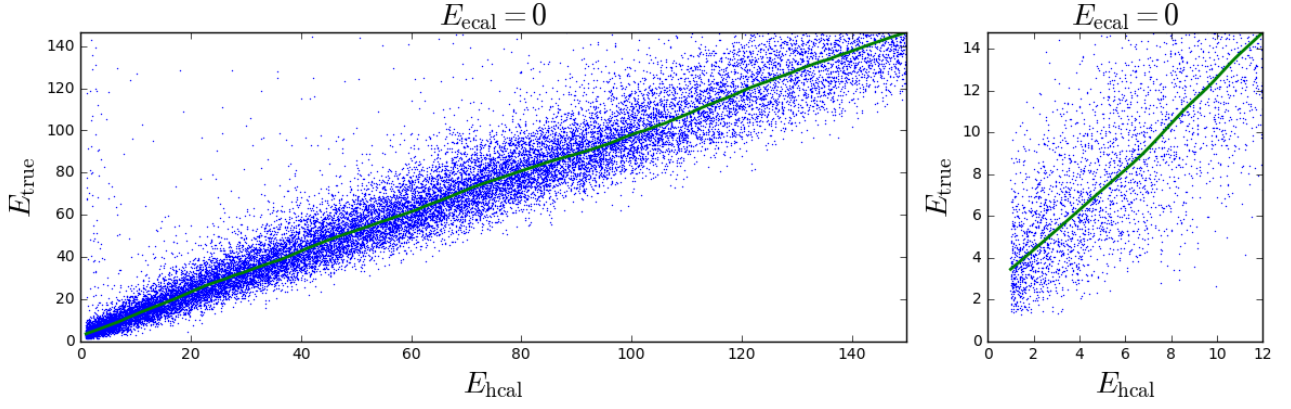
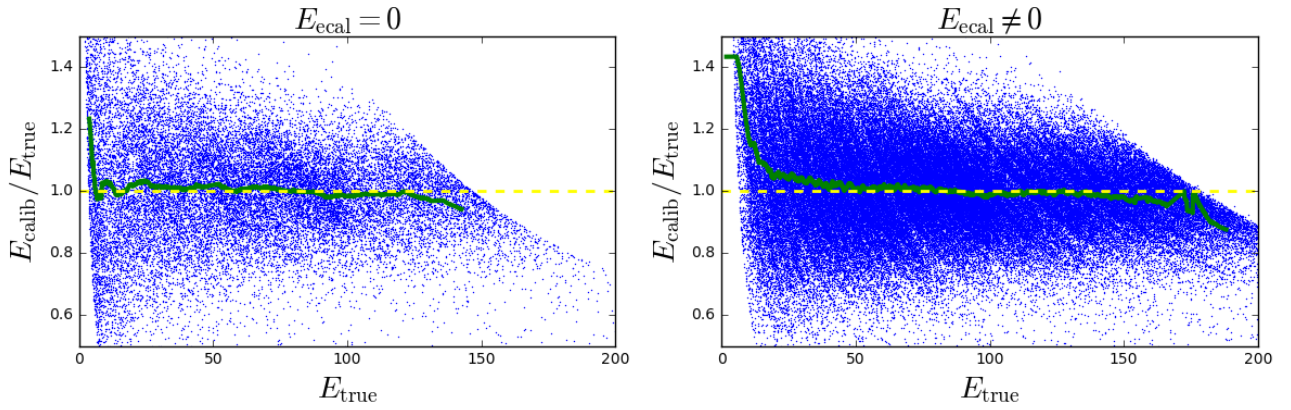
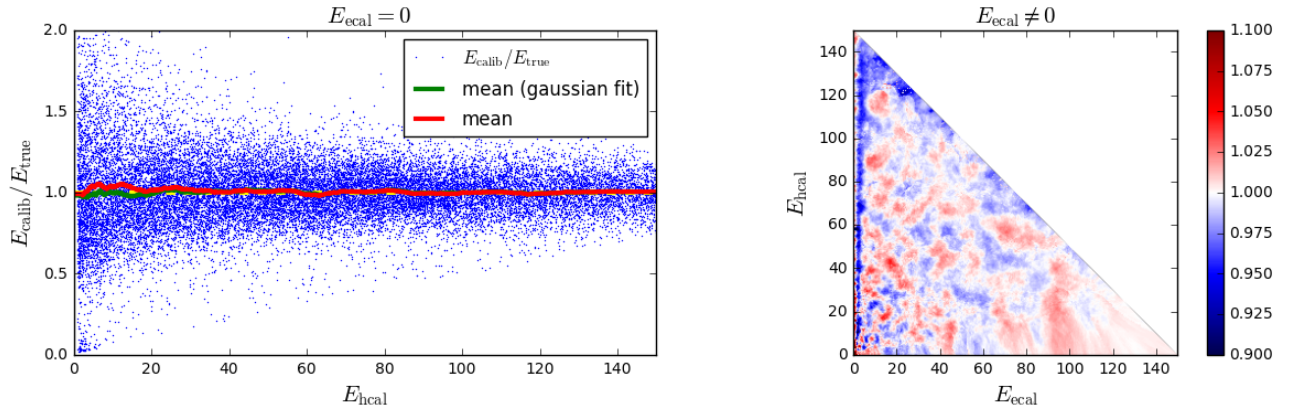


FIGURE 20 – Surface de calibration.

Effectuons alors la calibration d'un second jeu de données :



FIGURE 21 – Courbe de calibration pour  $E_{\text{ecal}} = 0$ .FIGURE 22 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{true}}$ .FIGURE 23 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{ecal}}$  et  $E_{\text{hcal}}$ .

### 5.3 Fit gaussien

Ici, il s'agit du même principe que précédemment mais nous allons considérer que la valeur de  $E_{\text{calib}}$  est la moyenne de la gaussienne. Principe de l'algorithme :

- on considère des points  $(E_{\text{ecal}}^{0,j}, E_{\text{hcal}}^{0,j})$  où nous allons évaluer l'énergie calibrée.
- pour chaque  $(E_{\text{ecal}}^{0,j}, E_{\text{hcal}}^{0,j})$  :

- on recherche ses  $k$  plus proches voisins dans le plan  $(E_{\text{ecal}}, E_{\text{hcal}}) \rightarrow (E_{\text{ecal}}^i, E_{\text{hcal}}^i), i \in [1, \dots, k]$
- on trouve la gaussienne correspondante  $\rightarrow \sigma, \mu$
- $\rightarrow E_{\text{calib}}^0 = \mu$

— on effectue une interpolation pour donner une valeur d'énergie calibrée quelque soit  $(E_{\text{ecal}}^0, E_{\text{hcal}}^0)$

Construisons alors la calibration [15] à l'aide des paramètres suivants :

$$\begin{cases} \text{lim} & = 150 \\ n_{\text{neighbors}, E_{\text{ecal}}=0} & = 2000 \\ n_{\text{neighbors}, E_{\text{ecal}} \neq 0} & = 250 \\ \text{lim} & = 150 \\ \text{algorithm} & = \text{'auto'} \\ \text{energystep}_{E_{\text{ecal}}=0} & = 1 \\ \text{energystep}_{E_{\text{ecal}} \neq 0} & = 5 \\ \text{kind} & = \text{'cubic'} \end{cases}$$

Calibration surface

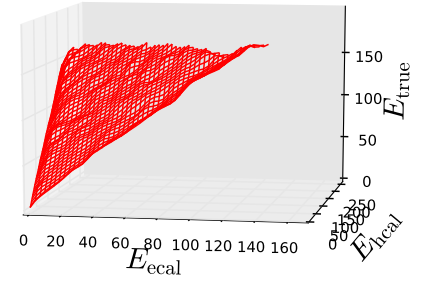
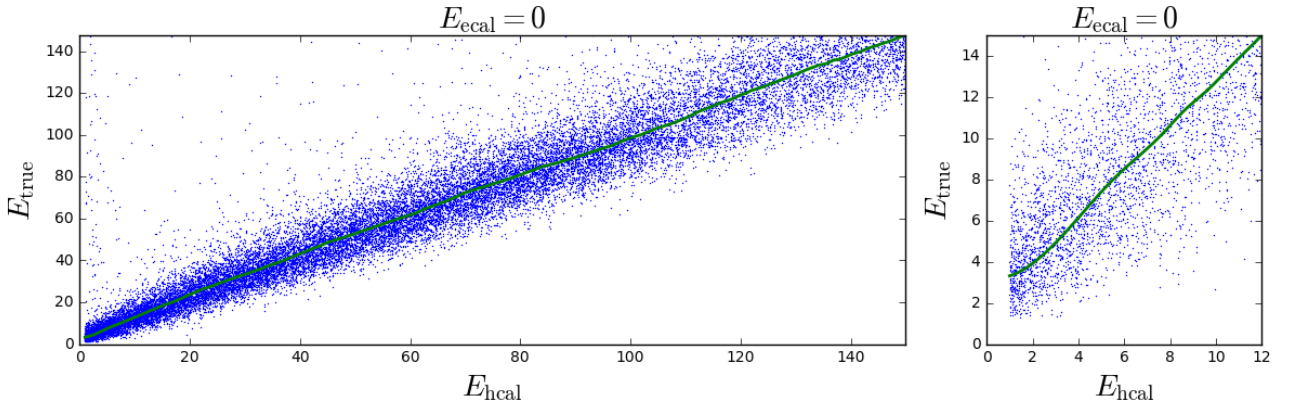
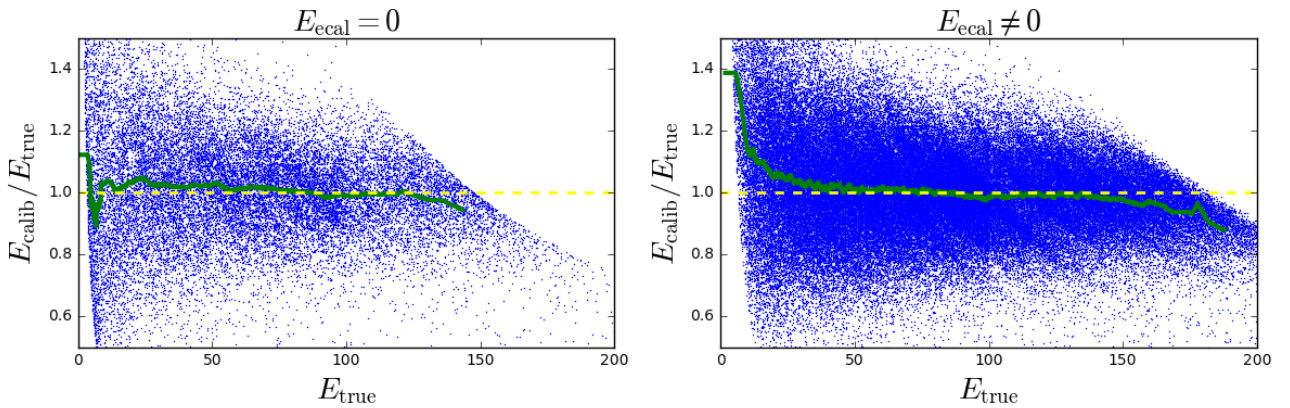


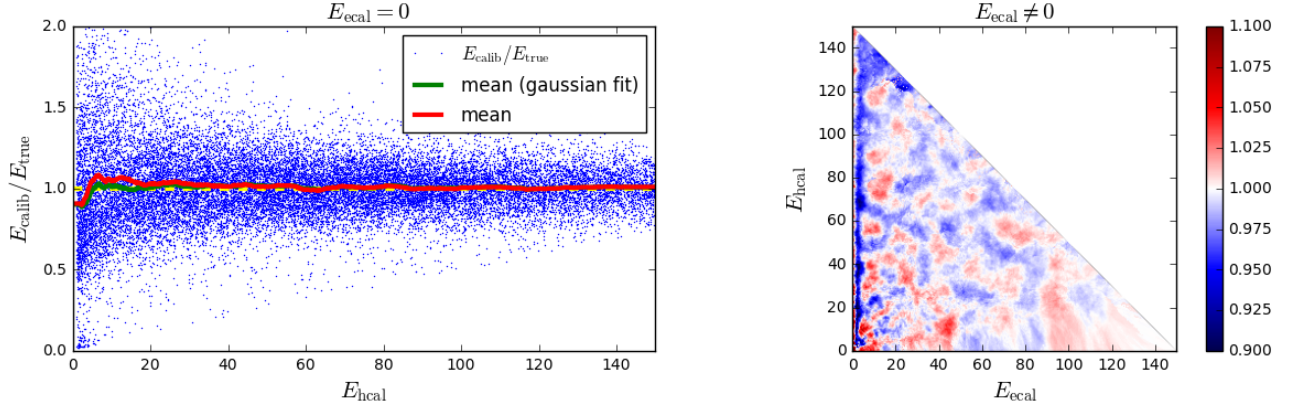
FIGURE 24 – Surface de calibration.

FIGURE 25 – Courbe de calibration pour  $E_{\text{ecal}} = 0$ .

Effectuons alors la calibration d'un second jeu de données :

FIGURE 26 –  $E_{\text{calib}}/E_{\text{true}}$  en fonction de  $E_{\text{true}}$ .



FIGURE 27 –  $E_{calib}/E_{true}$  en fonction de  $E_{ecal}$  et  $E_{hcal}$ .

## 6 Comparaison des méthodes

### 6.1 Méthodes des plus proches voisins

## 7 Annexes

### 7.1 Fonctions utiles du programme

DRAFT

## 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](https://sniang.github.io/particle_flow_calibration).
- [7] Scikit Learn. *Generalized Linear Models*. [scikit-learn.org](http://scikit-learn.org).
- [8] S. Niang. *Classe LinearRegression*. [https://github.com/sniang/particle\\_flow\\_calibration](https://github.com/sniang/particle_flow_calibration).
- [9] Scipy. *Optimization*. [docs.scipy.org](http://docs.scipy.org).
- [10] S. Niang. *Classe CalibrationLego*. [github.com/sniang/particle\\_flow\\_calibration](https://github.com/sniang/particle_flow_calibration).
- [11] Scikit Learn. *Nearest Neighbors*. [scikit-learn.org](http://scikit-learn.org).
- [12] S. Niang. *Classe KNN*. [github.com/sniang/particle\\_flow\\_calibration](https://github.com/sniang/particle_flow_calibration).
- [13] Scipy. *Interpolation*. [docs.scipy.org](http://docs.scipy.org).
- [14] S. Niang. *Classe KNNGC*. [github.com/sniang/particle\\_flow\\_calibration](https://github.com/sniang/particle_flow_calibration).
- [15] S. Niang. *Classe KNNGF*. [github.com/sniang/particle\\_flow\\_calibration](https://github.com/sniang/particle_flow_calibration).