

UNIVERSITÉ DE FRANCHE-COMTÉ



---

## Rapport Technique Photoelectric effect on interstellar grain

---

*Auteur :*  
NICOLAS BELLEMONT

*Référent :*  
JULIEN MONTILLAUD  
*Date de fin de développement :*  
VERSION GOLD : 15/10/2018  
*Format du code :*  
CODE SOURCE À EXÉCUTER  
*Language de programmation :*  
PYTHON3

## I. Liste des fichiers

- Simulation.py : fichier principal, à exécuter pour lancer les simulations.
- Functions.py : fichier contenant toutes les fonctions utilisés par le programme.
- Constant.py : fichier contenant les valeurs des constantes nécessaire au programme.
- Dépôt GitHub : <https://github.com/biolyse/Photoelectric-effect-on-interstellar-grain>

## II. Spécification fonctionnelle du programme

### II.1. But du programme

Le programme présenté dans ce rapport a pour but d'étudier l'effet de chauffage photoélectrique dû aux grains de poussière interstellaire. Ce phénomène se produit lorsqu'un photon de type UV est absorbé par un grain de poussière et que cette absorption déclenche l'éjection d'un électron à grande vitesse. Par la suite l'électron va transférer son énergie cinétique au gaz par le biais de collisions et ainsi chauffer l'ensemble.

Le fonctionnement de ce programme se découpe en deux étapes. Pour commencer il y a génération d'une matrice de 0 et de 1 représentant le grain de poussière (les 1 et les 0 correspondant respectivement à des emplacements occupé et non occupé par le grain).

Une fois la matrice représentative générée, le programme simule l'arrivée d'un certains nombres de photon (par défaut 1000000) sur ce grain et les différents événement qui peuvent s'en suivre (absorption de photon, émission d'électron...).

Ce programme permet donc de simuler le phénomène de chauffage photoélectrique et ainsi d'observer la proportion d'électron libéré par ce processus de chauffage.

### II.2. Données d'entrées

- choice : entier, permet à l'utilisateur de choisir quelle simulation il veut effectuer.
- number\_of\_grain : entier, détermine combien de grain seront générés pour l'exploitation choisi.
- sigma\_dens : réel, représente la densité de la distribution du grain (en pratique sert à faire varier le degré de "fractalité" d'un grain).

### II.3. Sortie du programme

Le programme ressort une série d'histogrammes correspondant à la simulation choisi par l'utilisateur. Chaque histogramme représentent le nombre de photon-électron qui se sont échappés du grain pour les différents paramètres choisi, ainsi que les énergies correspondantes.

### II.4. Conditions spécifique d'utilisation

- Obligation d'avoir dans le répertoire de travail des dossiers nommés *Grain\_Files* et *Results\_Files* afin de stocker les grains générés ainsi que les fichiers *.txt* correspondants.
- Dysfonctionnement possible : Selon la géométrie du grain, il est possible qu'un photon-électron parcours, une distance plus grande que celle le séparant du bord de la matrice représentant le grain. Dans cette situation il survient une erreur de type Out Of Bond.

## III. Fonctionnement interne du programme

### III.1. Description des modèles physiques

Le grain de poussière interstellaire réduit à deux dimensions, est modélisé par une matrice de 0 et de 1. Les 1 représentent les zones de l'image occupés par le grain et les 0 les zones de vide. Pour ce programme nous considérerons qu'il est intégralement composé de carbone.

La géométrie du grain est contrôlé par le biais d'un paramètre qui permet d'obtenir des formes plus ou moins fractale.

Comme les simulations se font sur des grains aux propriétés géométriques obtenus aléatoirement, il a été décidé que tous les photons de la simulation arriveraient depuis le même coté de la matrice représentante, cela équivaut à dire que les photons arrivent sur le grain depuis toutes les directions de l'espace.

En ce qui concerne l'éjection des photon-électrons, dans le modèle utilisé, si ceux ci rencontrent sur leurs trajectoire un trou (un 0) on considère qu'il se sont échappés du grain. Il est possible que ce "trou" ne soit dû qu'à la géométrie du grain et que l'électron puisse à nouveau rentrer dans ce dernier et continuer sa course

mais la fonction permettant de prendre cela en compte ralentissait énormément le programme<sup>1</sup>. Il à donc été convenue que si l'électron rencontrait un 0 sur son chemin il s'échappait du grain.

En ce qui concerne les énergies, si un électron à réussi à s'échapper du grain l'énergie cinétique qui lui reste est inscrite dans le fichier de résultat. Si le photon n'a pas touché le grain ou s'il n'est pas absorbé, la valeur -1 est rentré dans le fichier, si un électron n'est pas éjecté suite à l'absorption c'est la valeur -2 et si le photon-électron s'est fait ré-absorbé dans le grain c'est -3. Ces valeurs nous permettent de voir la proportion des photons qui n'ont pas participé à l'effet photoélectrique et à quel stade ils se sont arrêtés.

Les travaux de Bakes & Tielens<sup>2</sup> sont à la base des données numériques utilisées dans les simulations.

#### **III.2. Liste des éléments constitutif du programme**

Sera explicité dans cette partie le fichier principal ainsi que deux fonctions très importantes, pour le reste veuillez consulter les doc-strings présente en en-tête de chaque fonctions.

##### Simulation.py :

Demande à l'utilisateur de choisir une simulation à réaliser, puis les paramètres avec lesquels réaliser la simulation en question.

Par la suite un grain est généré sur la base des paramètres rentrés. Cela signifie que n'importe quel nombre de grains de n'importe quelle combinaison de paramètres peut être crée puis analysés.

##### GenGrain :

GenGrain est une fonction qui permet de générer une matrice de 0 et de 1 représentant un grain interstellaire. Elle prend en entrée les variables suivantes :

- sigma\_dens : réel, représente la densité de la distribution du grain
- number\_of\_grain : entier, détermine combien de grain seront générés pour l'exploitation choisi.
- test : booléen, permet de savoir sous quel format enregistrer le fichier contenant la matrice

Cette fonction provient initialement d'un programme donné par le professeur référent<sup>3</sup>. Le code à été intégralement traduit de python2 vers python3 puis adapté afin de devenir une fonction.

##### Main :

Main est une fonction qui simule l'arrivée de plusieurs photons sur le grain puis leurs comportement à l'intérieur de celui-ci. Elle prend en entrée les variables suivantes :

- number\_of\_grain : entier, détermine combien de grain seront générés pour l'exploitation choisi.
- sigma\_dens : réel, représente la densité de la distribution du grain (en pratique sers à faire varier le degré de "fractalité" d'un grain).
- GRAIN\_RADIUS : réel, taille effective du grain interstellaire en mètres.
- choice : entier, choix de la simulation à effectuer

Cette fonction à un comportement légèrement différent en fonctions de la simulation choisi. Pour les simulations 1 et 4 comme les grains ont les mêmes propriétés fractale et que le nom des fichiers est généré en fonction de ces propriétés, la fonction traite des fichiers dont le nom se termine par un \_i où i est le numéro du *i<sup>ème</sup>* grain généré. Ormis cette différence d'appellation, la fonction à le même comportement peut importe la simulation choisie [1]

##### Données et variables internes :

- absorption\_column : entier, colonne dans laquelle le photon est absorbé.
- angle : entier, angle d'éjection (en dégrée) du photon électron par rapport à la direction d'arrivé du photon.
- contact\_pixel : entier, colonne dans lequel le photon à rencontré le grain.
- da : réel, distance parcouru par le photon à l'intérieur du grain.
- de : réel, distance parcouru par le photon électron.
- energy : réel, valeur de l'énergie du photon incident.
- dim1/dim2 : entier, dimension de la matrice du grain.

---

1. Cela correspond à la fonction commenté freedom dans le fichier de fonction

2. The Photoelectric heating mechanism for very small graphitic grains and polycyclic aromatic hydrocarbons

3. Mr. Julien Montillaud

- GRAIN\_SIZE : réel, taille du grain en mètre.
- is\_absorbed : booléen, si le photon a été absorbé ou non.
- is\_ejected : booléen, si l'électron est éjecté ou non.
- is\_free : booléen, si l'électron est sorti du grain ou non.
- kinetic\_energy : réel, énergie cinétique de l'électron après être sorti du grain.
- matrix : matrice, matrice représentante du grain.
- p : entier, partie décimale du paramètre sigma.
- parameter : réel, paramètre du générateur non uniforme suivant une loi exponentielle.
- photon\_init\_position : entier, ligne de la matrice sur laquelle arrive le photon.
- pixel\_size : réel, taille d'un "pixel" (0 ou 1) en mètre.
- row\_photon : matrice, vecteur contenant la ligne d'arrivée du photon.
- S : entier, partie entière du paramètre sigma.

### III.3. Dépendance à des fonctions, objet issus de bibliothèques externe

Pour ce programme les bibliothèques suivantes ont été utilisées :

- Numpy
- Matplotlib
- Random
- Math

### III.4. Diagramme fonctionnel du programme

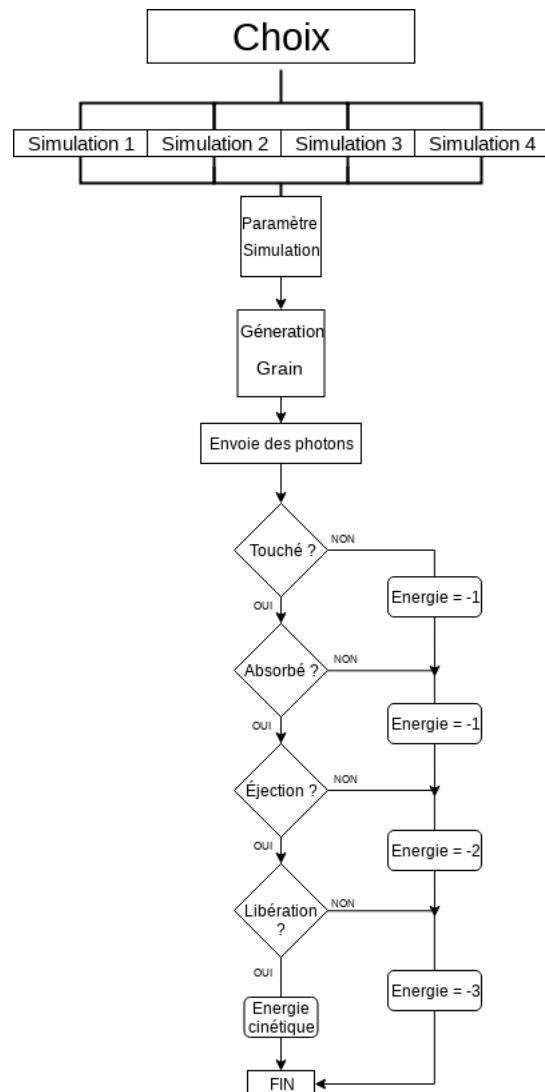


FIGURE 1 – Diagramme fonctionnel du programme

## IV. Fiabilité du programme, démarche qualité

La principale démarche qualité fut de comparer le code à ceux des personnes travaillant sur le même sujet afin de déceler des divergences et des potentielles erreurs de programmation.

Comme il n'y avait aucun résultat expérimentaux sur lesquels nous pouvions nous appuyer, la compréhension scientifique du phénomène photo-électrique relève plus de l'interprétation que de la certitude.

## V. Exemples de résultats du programme

### V.1. Quelques exemples de test réalisé avec le code

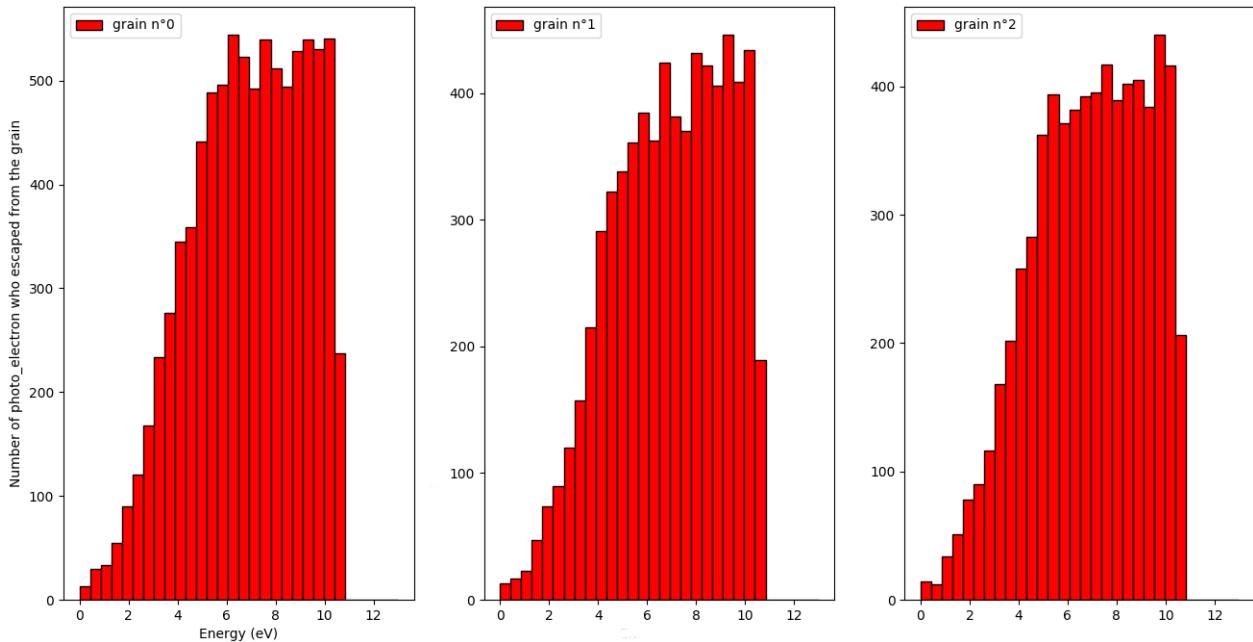


FIGURE 2 – Première simulation : 3 grains simulés avec le même paramètre,  $\sigma = 0.4$

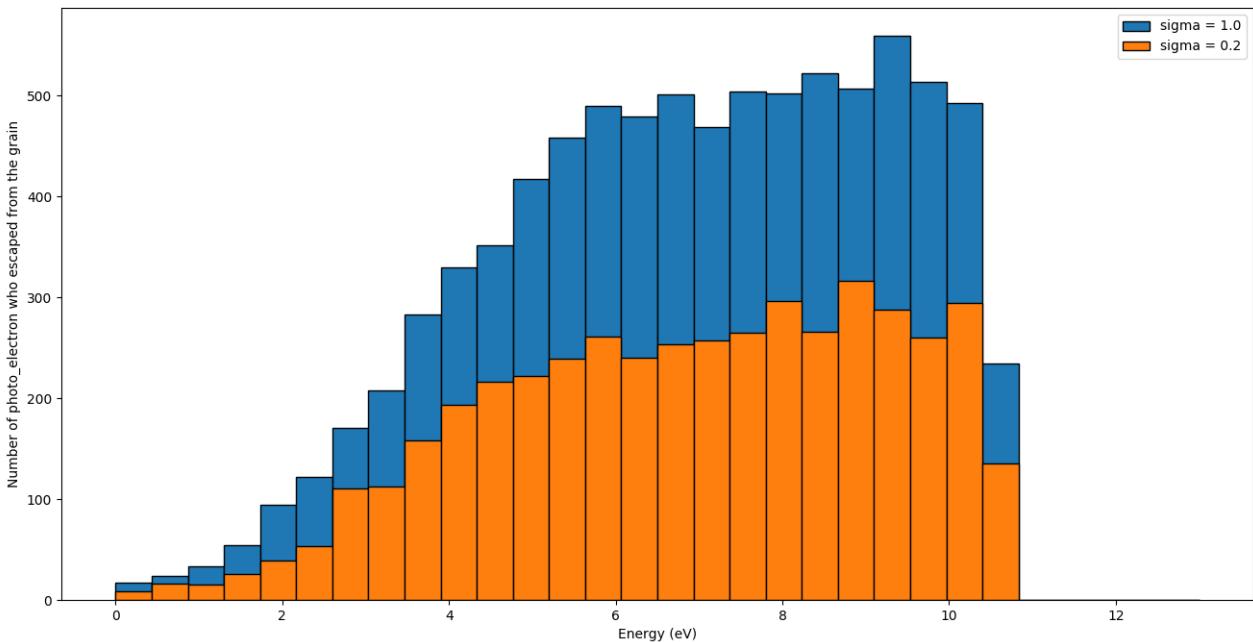


FIGURE 3 – Deuxième simulation : Un grain fractale ( $\sigma = 1.0$ ) et un grain sphérique ( $\sigma = 0.2$ )

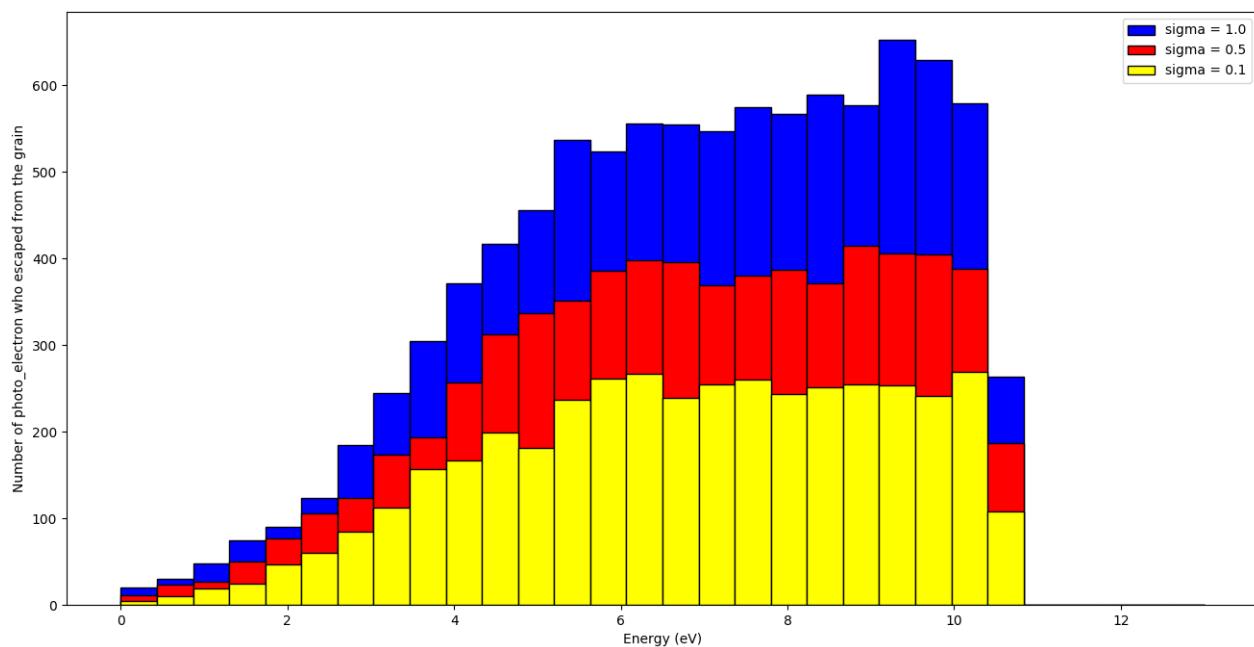


FIGURE 4 – Troisième simulation : 3 grains avec différents paramètres de fractalité

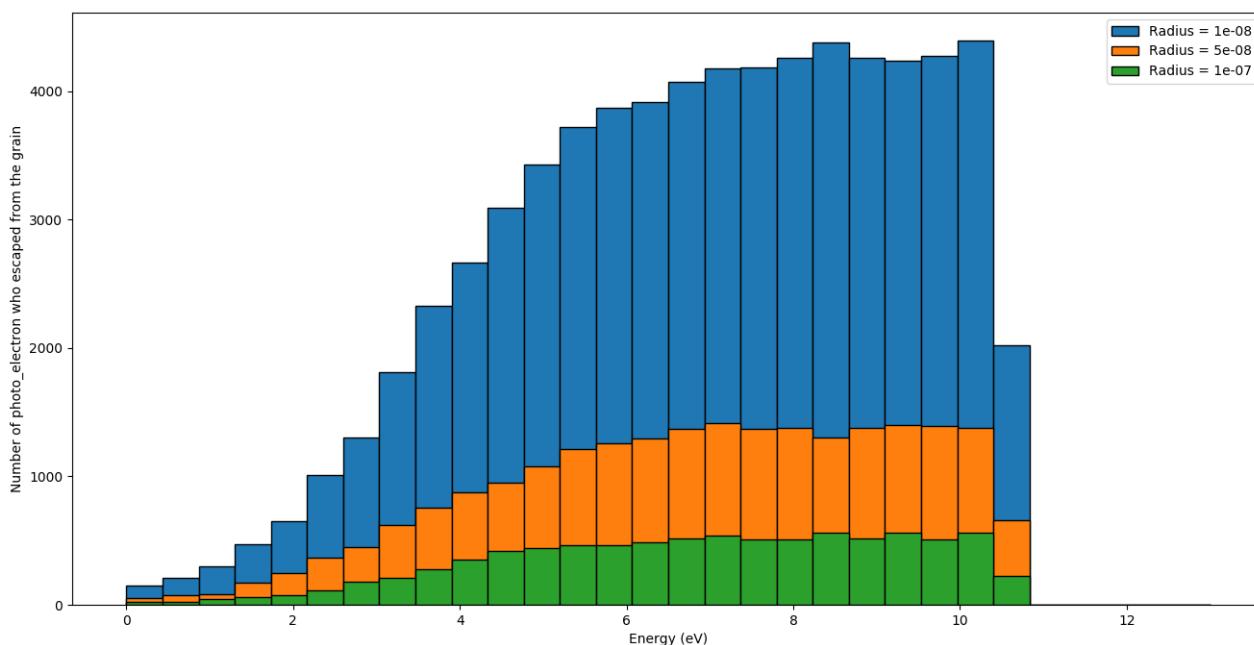


FIGURE 5 – Quatrième simulation : 3 différentes tailles de grain

## V.2. Interprétation physiques des résultats

La répétitions des différentes simulations à permis d'en arriver aux conclusions suivantes :

- Lorsque plusieurs grains sont générés à partir des mêmes propriétés fractales, seule la géométrie "visuelle" diffère. Cela se traduit par le fait que le nombre de photon-électron qui se sont échappés est du même ordre de grandeur pour chaque grain [2]. Le phénomène photoélectrique ne dépend pas de la variation géométrique du grain.
- Plus un grain est fractale plus il permet l'échappement de photon-électron [3][4]. Cela est dû au fait que la structure fractale permet plus facilement à un électron de d'atteindre un potentiel bord du grain et donc de s'en échapper.  
A contrario si le grain tend à être sphérique, peut importe où l'électron est émis il devra, dans la majeur partie du temps, traverser l'entièreté du grain, de ce fait il voit la probabilité de se faire ré-absorbé augmenter de manière significative.
- Plus la taille effective du grain est grande et plus la probabilité que l'électron se fasse absorbé est grande [5]. En effet peu importe la géométrie du grain (fractale ou sphérique) étant donné que l'électron doit parcourir une certaine distance une fois émis, si le grain deviens trop grand l'électron se fera absorbé.

Pour conclure il ressort que le phénomène photoélectrique est intimement corrélé à deux choses. Premièrement à la complexité de la géométrie du grain, un grain fractale favorisera plus l'effet photoélectrique qu'un grain sphérique. Deuxièmement à la taille du grain, plus celui ci sera grand et moins l'effet photoélectrique se manifestera.

