# Présentation

QMandelbrotCuda3D est un logiciel de calcul de fractals à partir des quaternions. Avec quelques paramètres, il crée un nuage de point. Un traitement avec le scripte txt2stl permet de créer le maillage et ainsi obtenir un objet 3D.

# Prérequis :

Il y a des prérequis pour utiliser QMandelbrotCuda3D.

## Logiciel / OS :

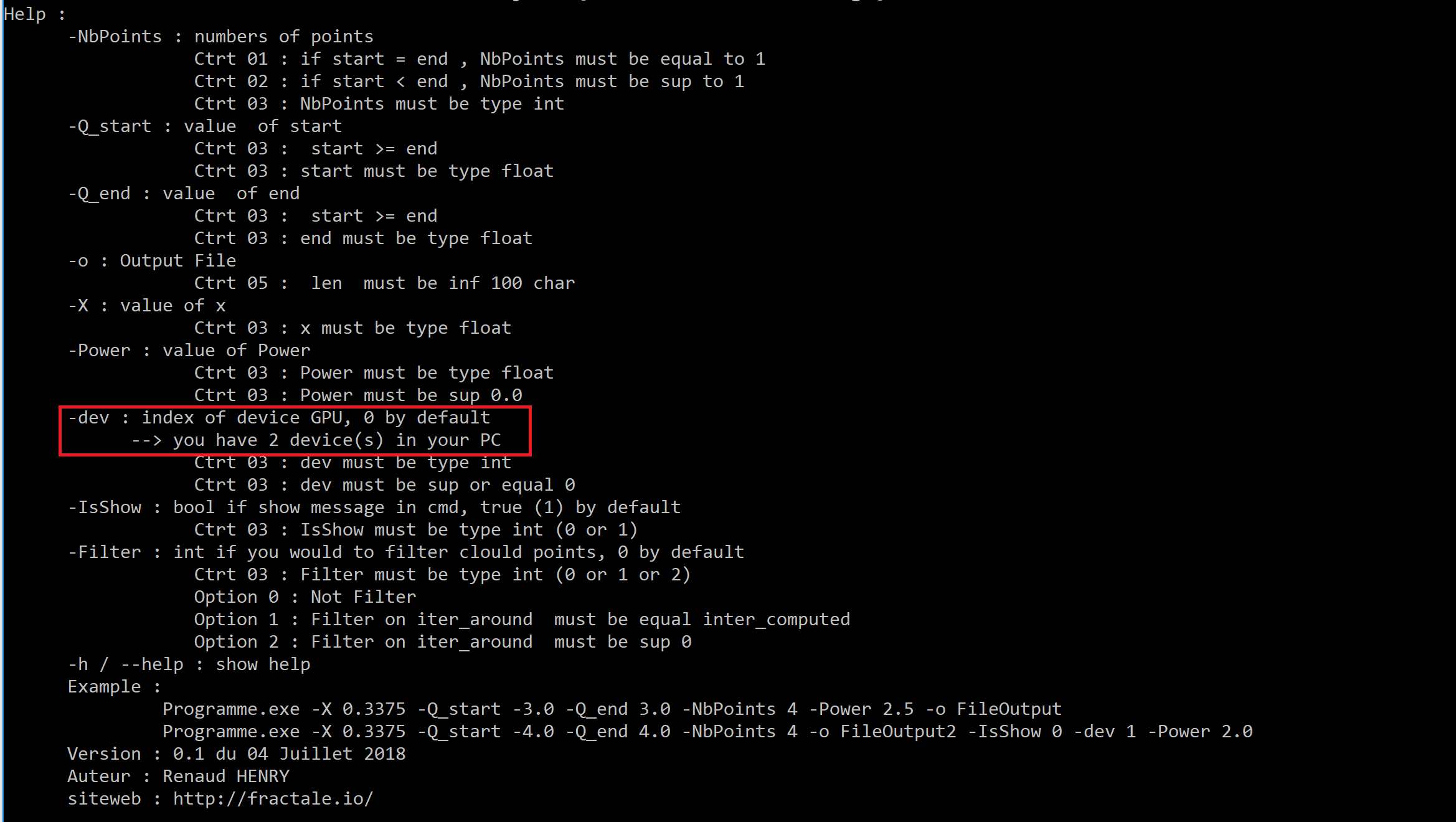
QMandelbrotCuda3D est programmé sur W10 avec les dernières mise à jours. Le code est compatible Linux mais avec des adaptions.

## Matériel :

QMandelbrotCuda3D est en cuda, donc uniquement compatible avec des cartes graphique nvidia Kepler ou Pascal afin de bénéficier des fonctionnalités de la mémoire Unifiée (Unified Memory).

De même, je conseil d’avoir deux cartes graphiques, une pour les applications et l’autre pour les calculs intensifs.

Si vous ne savez pas combien de carte graphiques sont détectables, lancez le scripte Help.cmd.



## Logiciels complémentaires (non indispensable)

Vous avez besoin de deux logiciels complémentaires :

* CloudCompare : il permet de visualiser le fichier txt généré par QMandelbrotCuda3D
  + Il est conseillé pour afficher les fractales non filtrées, car il accepte des fichiers avec 2 milliards de point maximum.
  + Site officiel : [www.danielgm.net/cc/](http://www.danielgm.net/cc/)
* MeshLab : il permet de faire le maillage, le scripte txt2stl utilise plus précisément le serveur MeshLab
  + Il est conseillé pour maillé les fractales filtrées, car l’affichage est limité en terme de nombres de point.
  + Site officiel : <http://www.meshlab.net/>

# Mode d'emploi :

## Installation de QMandelbrotCuda3D

Décompresser l’archive dans le dossier QMandelbrotCuda3DV\_0.1

## Installation des logiciels complémentaires

Allez sur le site de CloudCompare et de MeshLab  pour les installer.

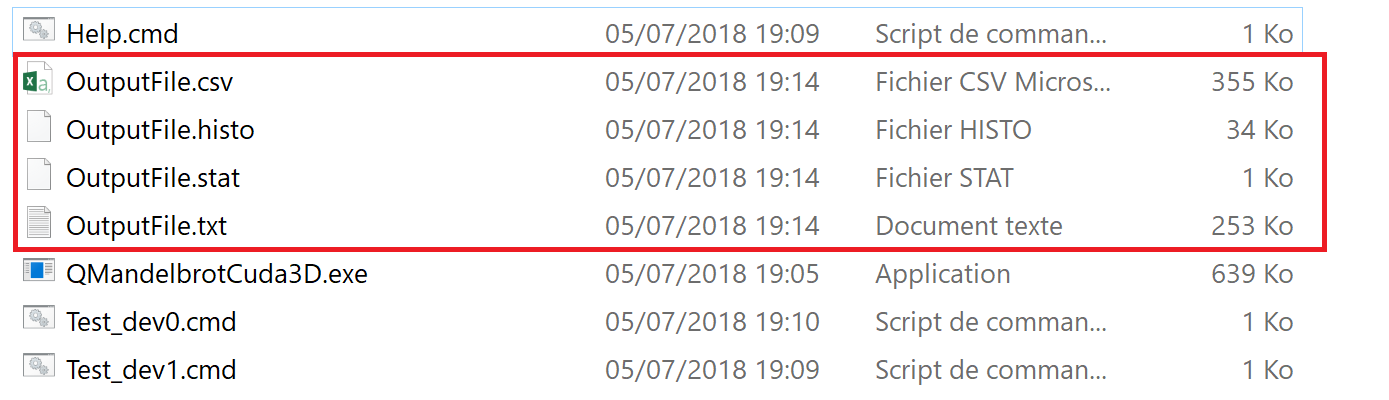
## Phase de test de QMandelbrotCuda3D

### Génération des données

Afin de tester l’installation, lancer le scripte :

* Test\_dev0.cmd si vous avez une seule carte graphique
* Test\_dev1.cmd si vous avez deux cartes graphiques

Le test consiste à faire une simulation rapide afin d’obtenir ceci :

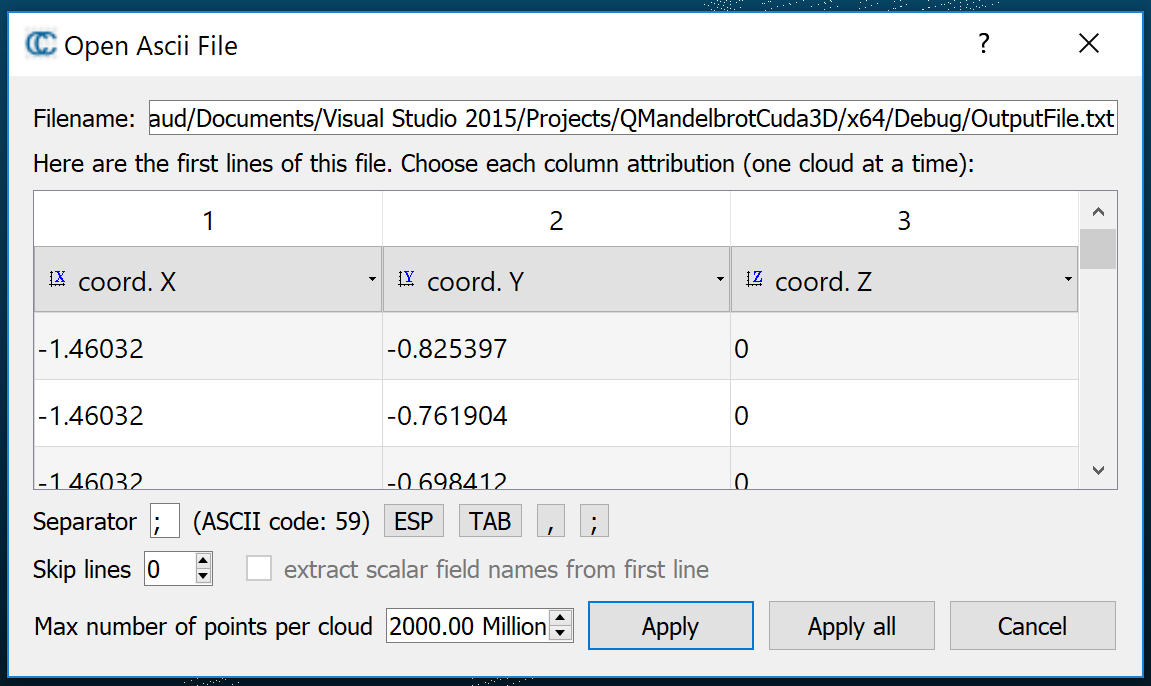
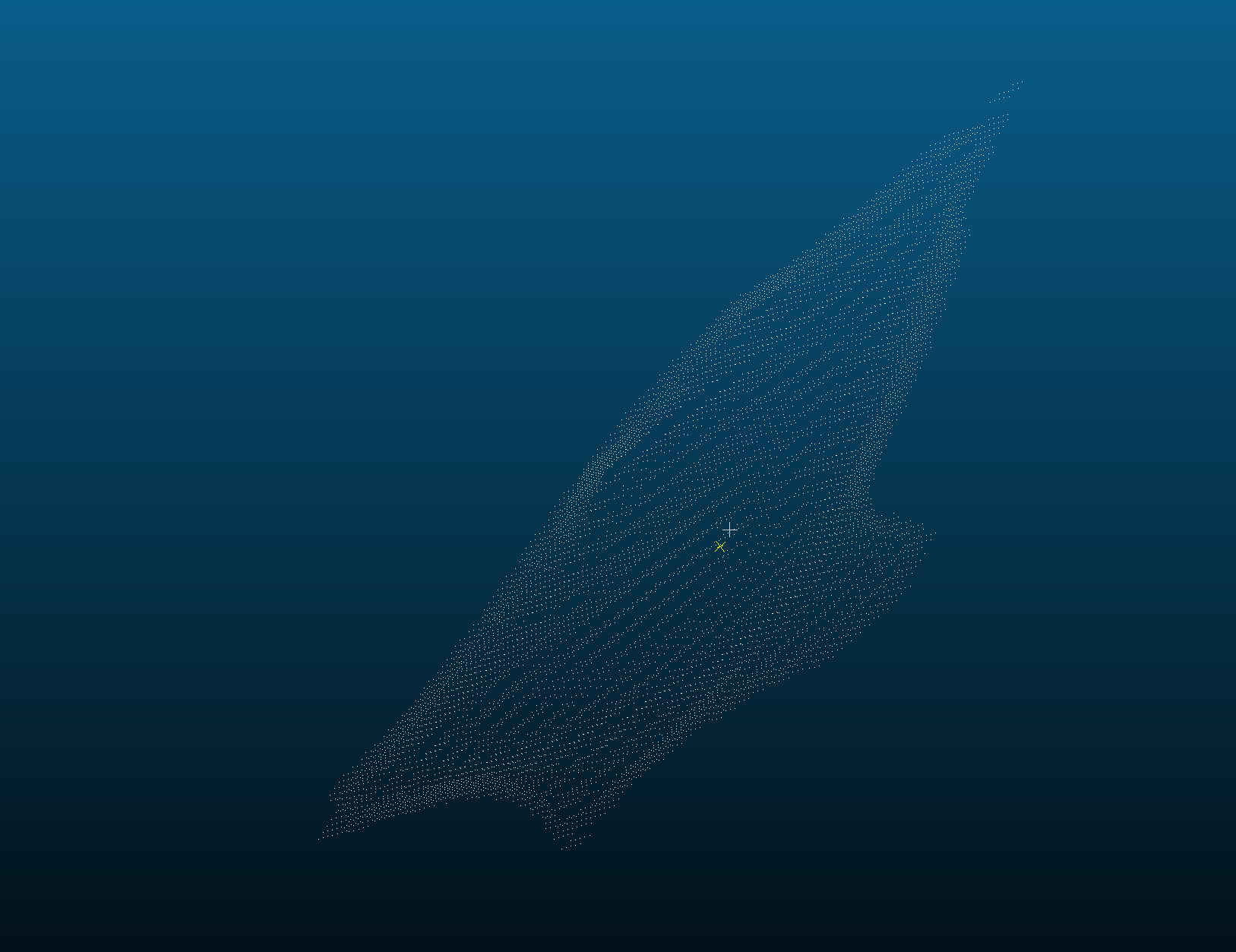


Vous obtenez quatre fichiers avec la même Racine « OutputFile », mais avec 4 extensions différentes :

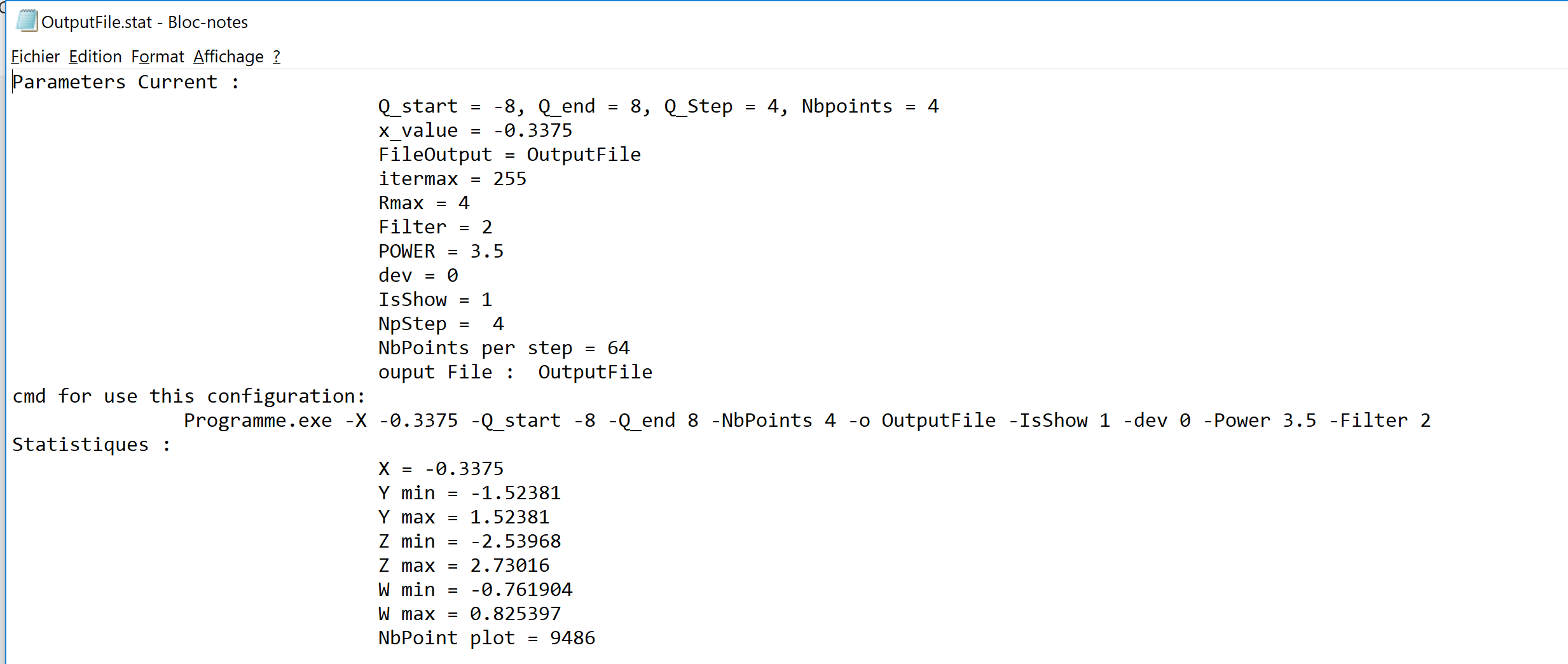
* .csv : Fichier tableur, s’ouvre avec Excel. Il contient la liste des points et l’itération correspondante. Attention sans filtre, le fichier peut dépasser rapidement le million de ligne (limite de Excel)
* .histo : histogramme des points, permet de connaitre la répartition des points en fonction de leurs itération maximale. Il s’ouvre avec Excel.
* .stat : statistique de la simulation, il synthétise :
  + Les paramètres de la simulation.
  + La ligne de commande à lancer pour réexécuter la simulation.
  + Les points limites (min, max) ayant une itération supérieure à 0.
* .txt : Fichier de la liste des points (X,Y,Z), utiliser pour la visualisation et le traitement

### Visualisation des données

#### Aperçu 3D de la fractale avec CloudCompare

1. Lancer CloudCompare
2. Ouvrir le fichier « OutputFile.txt »
   * File -->Open -->OutputFile.txt
   * Cliquer sur Open
3. Une fenêtre de configuration apparait, cliquer sur « Apply »
   * 
4. Vous pouvez maintenant visualiser la forme du contour de la fractale.
   * 

#### Analyse des statistiques :

1. Ouvrir le fichier « OutputFile.stat » avec le Bloc-notes
2. Vous devriez voir cet écran :
   * 
3. Dans la partie « Parameter Current », vous avez tous les paramètres utilisé pour la simulation, même ceux qui sont fixés en dure dans le programme, comme par exemple itermax,Rmax …
4. Dans la seconde partie, vous avez la ligne de commande avec les arguments pour reproduire la simulation.
5. Dans la dernière partie vous avec les limites mini et maxi des points de la fractale.

## Maillage avec « MeshLab »  :

Avoir un nuage de points c’est bien, mais des surfaces, c’est mieux.

### Prérequis :

Il faut avoir installé « MeshLab »

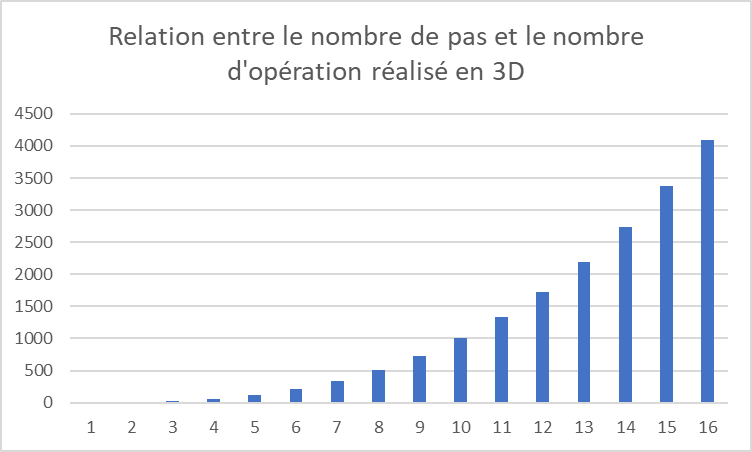
### Mode opératoire

1. Copier le fichier \*.txt dans le répertoire « Input » du programme « txt2stl »
   * **Attention : il ne doit pas avoir d’espace dans le nom du fichier.**
2. Lancer la commande « txt2stl.cmd »
3. Récupérez le fichier stl dans le dossier « Output »
4. Ouvrir le Fichier stl soit avec W10, soit avec MeshLab

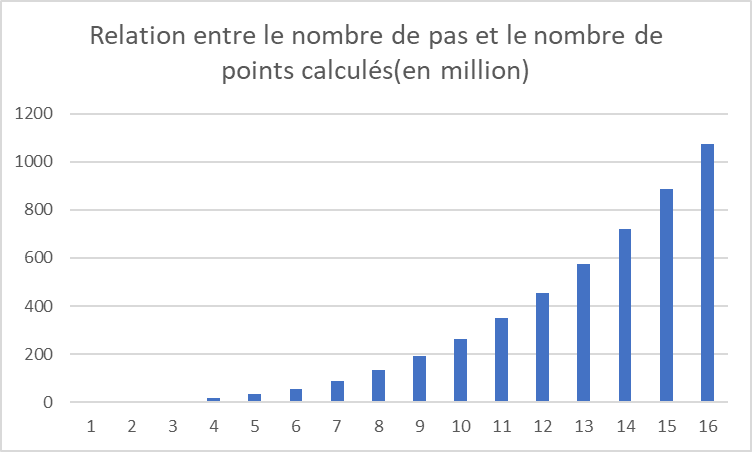
# Utilisation des paramètres :

Il y a neuf paramètres qui permettent de configurer une simulation de QMandelbrotCuda3D :

* -X :
  + Comme un quaternion est composé de quatre paramètres (x, y, z, w) et que seuls les objets en 3D sont modélisable, il faut fixer un paramètre. Donc dans une simulation « x » est fixe.
  + Contrainte(s) :
    - Il doit être un float, un nombre à virgule
    - Il doit être compris entre -20.00 et 20.00
* -Q\_start  et -Q\_end:
  + Comme le paramètre « x » est fixe, les autres paramètres (y,z,w) sont variables entre deux bornes, Q\_start et Q\_end.
  + Il faut que Q\_start soit inférieur à Q\_end.
  + Q\_start et Q\_end accepte les float, les nombres à virgule.
  + Les valeurs conseillées sont :
    - Q\_start=-8.0
    - Q\_end = 8.0
* - NbPoints :
  + Le nombre de points est en faites le nombre de pas sur une dimension. Donc il faut élever au cube pour avoir le nombre d’opération réalisée :



* + Il faut savoir aussi qu’a chaque pas, un cube de 643points est calculé, donc 262 144 points par pas.



* + Il est conseillé de se limité à 10 pas sur une dimension. Cela fait 262 millions de points calculer (sans compter les filtres)
* -Filter :
  + Il y a 3 options pour les filtres :
    - Filter =0 : Il n’y a aucun filtre, c’est le mode le plus économique en calcule, cependant le maillage est impossible à réaliser et le fichier des points est le plus volumineux.
    - Filter =1 : C’est le filtre qui ne laisse que les couches internétaires et d’enveloppe. Cependant le maillage est impossible sans un filtre (via le fichier « .csv ») et la taille du fichier des points est moyen.
    - Filter =2 : C’est le filtre qui ne laisse que la couche d’enveloppe. C’est seulement avec lui que le maillage est possible et que la taille du fichier des points est le plus petit.
* -IsShow :
  + Il y a 2 options pour l’affichage :
    - IsShow =0 : Seuls les messages qui informent le passage d’une étape à une autre est affiché.
    - IsShow =1 : tous les messages sont affichés
* -dev :
  + C’est l’indexe de la carte graphique utilisée. Il est conseillé de le mettre à 0. Par défaut il est à 1 (ma config).
* -Power :
  + C’est la puissance qui élève le quaternion. Il doit être supérieur ou égal à 2.0 et peut être un nombre à virgule. Comme le programme utilise la forme polaire du quaternion, quelque soit la puissance, le nombre d’opération est identique.
* -o :
  + Le fichier de sortie ne doit pas comporter d’espace ou de caractères interdits par windows.

# Réalisation d’une fractale :

Il est conseillé de faire un repérage avant de lancer un calcule gourmant en temps. Donc la génération se fait en 2 étapes. Une phase de prospection et une phase de rendu HD

## Etape 1 : prospection

Dans un premier temps il faut élargir le champs de recherche, c’est-à-dire mettre -Q\_start  et -Q\_end à -10.0 et 10.0 afin d’englober tous l’espace de recherche. Puis fixer à 2 ou 3 le nombre des Pas (-NbPoints), mettre à 0 le filtre pour gagner encore en temps de calcule. Après il reste à jouer avec deux paramètres, -X et -Power. Sans oublier de paramétrer -dev.

Exemple :

QMandelbrotCuda3D.exe -X 0.3375 -Q\_start -10.0 -Q\_end 10.0 -NbPoints 3 -dev 0 -Power 3.5 -Filter 0

Visualiser le résultat (ficher .txt) sur « CloudCompare ».

## Etape 2 : rendu HD

Une fois la forme fractale voulue obtenue, on va passer au rendu HD. Reprenons les paramètres -X 0.3375 et -Power 3.5. Avec le fichier Stat (\*.stat) on connait les limites min et max de la fractale :

* X = 0.3375
* Y min = -1.53439
* Y max = 1.53439
* Z min = -2.48677
* Z max = 2.48677
* W min = -0.687831
* W max = 0.793651

C’est le Z qui englobe les autres paramètres, donc la commande pour rendre la fractale en HD est :

QMandelbrotCuda3D.exe -X 0.3375 -Q\_start –2.55 -Q\_end 2.55 -NbPoints 10 -dev 0 -Power 3.5 -Filter 2 -IsShow 0

On obtient ainsi après 12 minutes de calcul et une étape de maillage :

