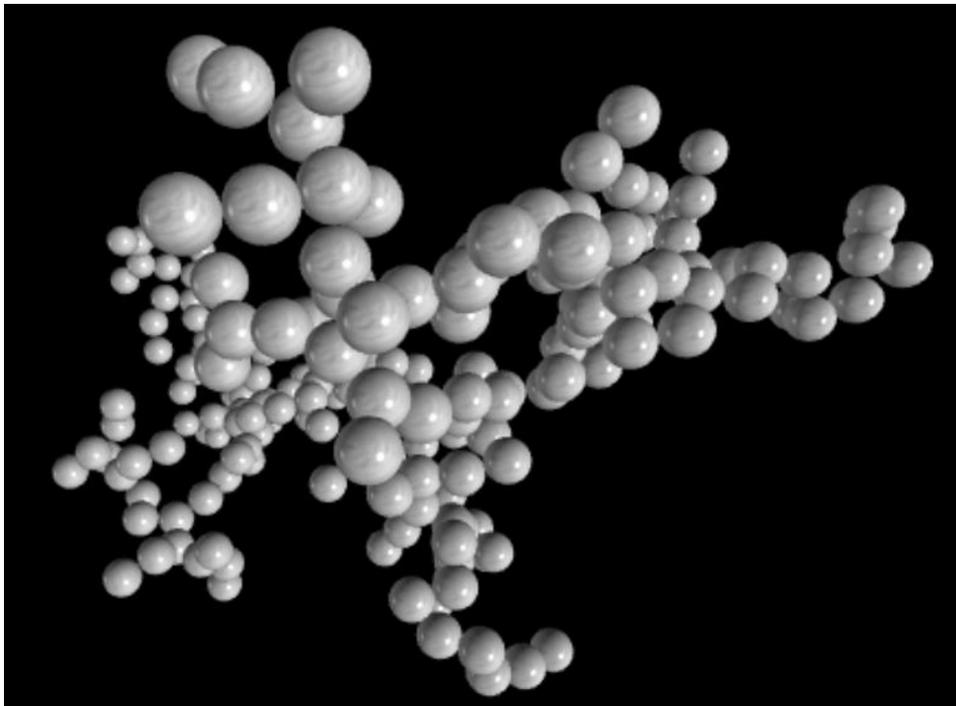


# **L'interaction de Van der Waals d'agglomérats de nanoparticules simplifiée**



***Etudiants :***

Paul Bismuth-Harhar

Doorvisha Sonali Mahabirsingh

Raphaël Schirru

Bastien Diot

Justine Saboureau

Juliette Visse

***Enseignant-responsable du projet :***

José Morán



**Date de remise du rapport :** 12/06/2021

**Référence du projet :** STPI/P6/2021 – 25

**Intitulé du projet :**

*L'interaction de Van der Waals d'agglomérats de nanoparticules simplifiée*

**Type de projet :** Bibliographie, Simulation Numérique

**Objectifs du projet :**

L'objectif de ce projet de physique est de simuler des agglomérats de nanoparticules à l'aide d'un logiciel de libre accès nommé FracVAL. Pour cela nous avons compris et modifié le code Python fourni par notre professeur référent permettant de modéliser le comportement de la constante de Hamaker dépendant du matériau utilisé, dans notre sujet le titania.

De plus, nous devons calculer les énergies d'interaction de Van der Waals entre des agglomérats de nanoparticules et entre leurs sphères de volume équivalent dans le but de pouvoir modéliser les courbes correspondantes.

Nous allons ensuite modifier les paramètres clés pour analyser et interpréter les différentes courbes obtenues. Finalement, nous voulons modéliser une courbe théorique correspondant à ce modèle.

**Mots-clés du projet :**

Agglomérats, Énergie, Van der Waals

## TABLE DES MATIERES

INTRODUCTION	6
1. MÉTHODOLOGIE, ORGANISATION DU TRAVAIL	7
2. DOMAINE DES NANOPARTICULES	8
2.1. Définitions des différents termes : nanoparticules, agglomérats	8
2.2. Les interactions de Van der Waals	8
3. SIMULATION NUMÉRIQUE ET ANALYSE	9
3.1. Simulation d'un agglomérat de nanoparticules de titania avec FracVAL	9
3.2. Exploitation des données générées : calcul de potentiels et énergies	11
3.2.1 Adaptation du code Python à l'étude d'un agglomérat de titania	11
3.2.2. Exploitation des résultats: modélisation des énergies et analyse des courbes	12
3.3. Modélisation et variations des paramètres clés	14
3.3.1. Modélisation	14
3.3.2 Variations des paramètres clés	15
3.3.2.1 Variation du nombre de rotations	15
3.3.2.2 Nombre de particules	17
3.3.2.3 Paramètre fractals ( $D_f$ et $k_f$ )	18
4. CONCLUSION ET PERSPECTIVES	21
5. BIBLIOGRAPHIE	21
6. ANNEXE	21

## NOTATIONS, ACRONYMES

Listes des notations que nous utiliserons tout au long de ce dossier notamment dans l'utilisation des formules :

$N$  : Nombre de particules dans l'agglomérat

$k_f$  : Préfacteur fractal

$D_g$  : Diamètre de giration

$D_p$  : Diamètre d'une particule

$D_{max}$  : Diamètre d'un agglomérat

$r_{max}$  : Rayon d'un agglomérat

$D_f$  : Dimension fractale de l'agglomérat

$h$  : Distance entre les deux extrémités de deux agglomérats

$d$  : Distance entre les centres de deux agglomérats

$A_{131}$  : Constante de Hamaker

$A_{131,s}$  : Constante de Hamaker prenant en compte la contribution statique des dipôles permanents

$A_{131,0}$  : Constante de Hamaker non retardée

$\varphi$  : Densité de conditionnement de l'agglomérat

$x$  : Dimension géométrique (nombre de composantes utiles pour décrire la géométrie)

$h_{cr}$  : Distance caractéristique du retard

$\sigma_{LJ}$  : Distance de potentiel équivalent nul

De plus, nous nous servirons de l'acronyme VdW pour signifier Van der Waals dans ce rapport et le terme particule sera employé pour désigner un monomère.

## INTRODUCTION

Tout au long de ce quatrième semestre à l'INSA de Rouen, nous avons réalisé un travail de groupe dans le cadre du projet physique. Nous nous sommes intéressés à l'interaction de Van der Waals dans les agglomérats de nanoparticules simplifiée. Notre groupe est composé de 6 étudiants de STPI2 : Paul Bismuth-Harhar, Bastien Diot, Doorvisha Sonali Mahabirsingh, Justine Saboureau, Raphaël Schirru et Juliette Visse.

Les nanoparticules, autrement dit particules de taille inférieure à 100 nm, sont de plus en plus présentes dans notre quotidien, notamment dans les produits commercialisés. Elles sont utilisées dans de nombreux domaines comme la médecine, la cosmétique ou même l'agroalimentaire. Par exemple, depuis des décennies, des nanoparticules de noir de carbone sont utilisées pour renforcer la gomme des pneus de voitures [2].

L'intérêt des nanoparticules réside dans leur petite taille qui leur confère des propriétés particulières. Ce phénomène est notamment constaté dans les particules de métal, qui ont la capacité de colorer le verre. Un exemple intéressant est celui de la Coupe de Lycurgue (*Fig. 1*) : le verre de la coupe est de couleur verte à la lumière du jour mais prend une couleur rouge quand elle est éclairée de l'intérieur. Ceci est dû à l'interaction de la lumière avec les nanoparticules d'or présentes dans le verre [2].



*Figure 1 : Coupe de Lycurgue*

L'utilisation des nanoparticules est donc répandue, même si ce domaine demeure largement inexploré.

Pour ce projet, nous nous intéresserons à des nanoparticules de titania. Nous les retrouvons notamment sous la forme de dioxyde de titane dans les crèmes solaires [2]. À nouveau, l'intérêt d'utiliser des nanoparticules est flagrant : lorsque leur taille est de l'ordre du micromètre, ces oxydes forment une couche de couleur blanche et opaque. Cependant, lorsqu'ils sont utilisés sous forme de nanoparticules, ils deviennent transparents tout en conservant leurs propriétés de protection UV.

Nous allons étudier des agglomérats de ces nanoparticules en commençant tout d'abord par définir les termes nécessaires à la compréhension du sujet ainsi que le principe d'interaction de VdW. Puis nous simulerons un agglomérat sur le logiciel FracVAL afin d'analyser et exploiter les courbes et les données obtenues. Enfin, nous modifierons les paramètres morphologiques clés de ce projet afin d'évaluer leur importance.

## 1. MÉTHODOLOGIE, ORGANISATION DU TRAVAIL

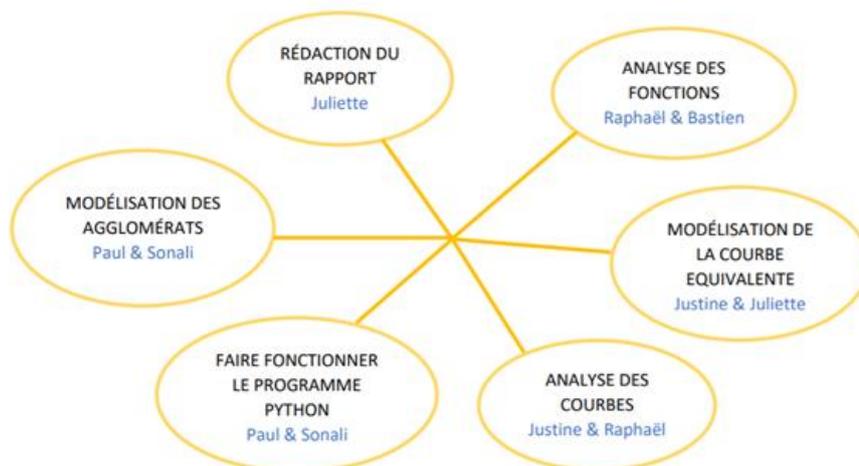
Malgré des débuts à distance rendant plus compliqué le travail collectif, nous nous sommes finalement rapidement retrouvés en présentiel. Nous avons pu nous rencontrer ainsi que notre professeur-référent M. José Morán qui nous a expliqué notre sujet, a répondu à nos questions et nous a accompagnés tout au long du semestre sur ce projet.

Pour réaliser au mieux ce projet, nous nous sommes répartis les tâches et organisés en sous-groupes dans le but d'obtenir un environnement autonome et efficace (Fig.2).

Tout d'abord, nous avons tous commencé par nous renseigner et prendre connaissance de notre sujet notamment en lisant le document de référence [1]. Juliette s'est occupée d'approfondir la recherche documentaire afin de déterminer les définitions ainsi que les formules du sujet. Paul et Sonali, possédant des ordinateurs suffisamment puissants, se sont occupés de télécharger et de faire fonctionner le logiciel FracVAL à l'aide d'un programme Python écrit par notre professeur qu'ils ont modifié. Ils ont ensuite pu approfondir le travail en modélisant les différents agglomérats ainsi que les courbes associées en fonction des différents paramètres. Ainsi, Justine et Raphaël ont pu compléter ce travail en analysant et comparant les différentes courbes. Un autre groupe composé de Bastien et Raphaël s'est occupé d'analyser le programme et les fonctions les plus importantes pour nous faciliter la lecture du programme. Pour compléter le travail, le dernier groupe avec Justine et Juliette, s'est concentré sur la modélisation de la courbe équivalente.

Même si nous étions répartis en sous-groupes pour la réalisation du travail, il fallait bien souvent se concerter et s'entraider pour avancer. De plus, tout notre travail effectué en autonomie était partagé avec le reste du groupe sur un drive commun nous permettant de suivre l'avancée du projet. Nous avons aussi mis en place un serveur Discord pour communiquer.

En complément, nous nous retrouvions chaque semaine à l'INSA avec notre professeur pour mettre en commun notre travail de la semaine, suivre l'avancement de notre projet, poser des questions et se répartir les tâches pour la semaine suivante.



*Figure 2 : Diagramme des tâches réalisées*

## 2. DOMAINE DES NANOPARTICULES

### 2.1. Définitions des différents termes : nanoparticules, agglomérats

Une nanoparticule est une particule de petite taille (entre 1 et 100 nm). Leur usage est relativement récent, depuis 1990 seulement, et est en pleine expansion. Pour commencer notre étude des nanoparticules, nous allons tout d'abord mettre en place quelques définitions importantes à la bonne compréhension du sujet.

En premier lieu, un agglomérat est un ensemble de nanoparticules, dont la configuration et la structure déterminent ses propriétés physiques et ses interactions avec le milieu extérieur, à la différence de l'agrégat qui est lié par des forces chimiques. Nous parlerons dans ce rapport uniquement d'agglomérat dont la structure dépend de nombreux facteurs comme les forces entre chaque particule. Nous étudierons ici seulement des agglomérats de type fractal. Une dimension fractale est caractérisée par la capacité de remplissage de l'espace.

Ce type d'agglomérat possède deux caractéristiques principales. La première concerne le nombre de particules appartenant à l'agglomérat qui suit la loi suivante :

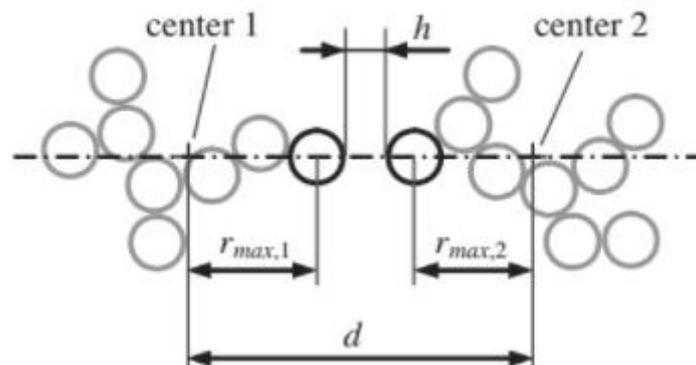
$$N = k_f \left( \frac{D_{max}}{D_p} \right)^{D_f} \text{ (Eq. 1)}$$

Cette équation considère que  $N$  augmente selon une loi de puissance avec une taille caractéristique de l'agglomérat  $\frac{D_{max}}{D_p}$ , ce qui est tout à fait en accord avec les particules d'oxyde de titane que nous allons étudier.

La seconde caractéristique se base sur la similitude de la structure à toutes les échelles. Comme dit précédemment, les agglomérats interagissent entre eux, notamment avec les forces de Van der Waals.

### 2.2. Les interactions de Van der Waals

Les interactions de Van der Waals se traduisent par une forte interaction entre des atomes ou des molécules situés à courte distance.



*Figure 3 : Schéma représentant une modélisation de l'interaction de Van der Waals entre deux agglomérats de nanoparticules sphériques à courte distance [1]*

Les interactions entre les agglomérats sont distinguées par deux situations limites en fonction de la distance entre ces derniers. Dans un premier temps, lorsque les agglomérats sont situés à grande distance, nous pouvons les considérer comme sphériques, et au contraire, s'ils sont très proches nous pouvons prendre uniquement en considération les interactions de VdW entre la paire de nanoparticules primaires les plus proches (*Fig.3*). Dans ce dernier cas, le centre de gravité des agglomérats est aligné sur une ligne droite, de même que les particules les plus extérieures qui, situés à une distance  $h$  l'une de l'autre, forment la paire de particules primaires.

Il existe aussi des agglomérats éloignés d'une distance intermédiaire pour lesquels nous ne pouvons pas négliger leur structure. Néanmoins, nous allons dans ce projet nous intéresser principalement aux interactions de VdW avec un modèle simplifié.

Ce modèle considère que l'interaction entre deux agglomérats peut être calculée en faisant la somme des interactions de chaque paire de nanoparticules. Pour cela nous composons cette paire d'une particule appartenant au premier agglomérat A et une particule du second agglomérat B. Effectivement, nous supposons que l'interaction de VdW entre une paire de nanoparticules est peu affectée par la présence des autres particules.

Nous obtenons donc la formule suivante, approximation de l'énergie de l'interaction de VdW entre deux groupes de particules : [1]

$$V_{vdW,A-B} = \sum \sum V_{vdW,i-j} \quad (Eq. 2)$$

Formule dans laquelle nous pouvons approcher la contribution par paire  $V_{vdW,i-j}$  avec la formule suivante (*Eq.3*) en prenant deux sphères de diamètre  $x_1$  et  $x_2$ . Ce calcul d'énergie d'interaction macroscopique est proposé par Hamaker :

$$V_{vdW} = -\frac{A_{131}}{12} \left[ \frac{x_1 x_2}{h \cdot (x_1 + x_2 + h)} + \frac{x_1 x_2}{(x_1 + h) \cdot (x_2 + h)} + 2 \ln \left( 1 - \frac{x_1 x_2}{(x_1 + h) \cdot (x_2 + h)} \right) \right] \quad (Eq. 3)$$

### 3. SIMULATION NUMÉRIQUE ET ANALYSE

#### 3.1. Simulation d'un agglomérat de nanoparticules de titania avec FracVAL

La première étape de la simulation numérique des interactions de VdW a consisté à générer un agglomérat de nanoparticules simplifié, en utilisant l'algorithme FracVAL. Ce code permet la simulation d'agglomérats similaires à des fractals, à partir desquels pourront être calculées des énergies d'interaction.

Différents paramètres définissant la structure de l'agglomérat sont préalablement entrés dans un fichier texte, tels que les paramètres fractals, le nombre de particules, ou encore la quantité d'agglomérats voulue (*Fig.4*).

```

module Ctes
  implicit none
  integer, parameter:: N=50                !Number of PP
  real, parameter:: Df=1.80                !Fractal dimension
  real, parameter:: kf=1.40                !Fractal prefactor
  real, parameter:: rp_g=15.0              !Geometric mean PP
  real, parameter:: rp_gstd=1.00           !Geometric PP standard deviation
  integer, parameter:: Quantity_aggregates = 1 !Quantity of aggregates to be generated
  integer, parameter:: Ext_case = 0        !Activate extreme cases
  real, parameter:: Nsubcl_perc = 0.1      !controls the subclusters size (keep alv
  real, parameter:: tol_ov=10.**(-6.)     !Tolerance to overlapping
  real, parameter :: pi=4.0*atan(1.0)
  real R(N)                                !Primary particles radii and mass
  real X(N), Y(N), Z(N)                    !Coordinates of primary particles
  integer:: iter, i, j, k
end module Ctes

```

*Figure 4 : fichier de paramètres nécessaires à la génération d'un agglomérat utilisant FracVAL*

Le code est ensuite exécuté à travers une interface de commandes, en utilisant un processus itératif. Des fichiers de type texte sont générés, ils contiennent deux informations: les coordonnées dans l'espace (O, x, y, z) de chaque particule de l'agglomérat, ainsi que son rayon (Fig.5).

102.377357	43.7041359	15.8481064	15.0000000
77.2057419	8.05399323	77.5793076	15.0000000
115.257721	28.0143776	-6.24099731	15.0000000
17.9422607	-19.9040184	80.5534973	15.0000000
27.2730083	6.48132324	69.7483597	15.0000000
2.18298340	-25.0193577	102.205467	15.0000000
50.1415367	0.134750366	51.3975754	15.0000000
18.1123581	-4.66054535	117.429588	15.0000000
14.6795082	-87.0125732	130.198120	15.0000000
32.4114189	-103.428078	147.977615	15.0000000
20.5725079	-76.4815979	102.732300	15.0000000
42.0142860	-131.679855	144.875626	15.0000000
Coordonnée x	Coordonnée y	Coordonnée z	Rayon

*Figure 5 : fichier de paramètres généré avec FracVAL*

Ce fichier nous servira de base pour calculer des énergies d'interactions au sein de l'agglomérat, en utilisant le code Python fourni par notre professeur.

### 3.2. Exploitation des données générées : calcul de potentiels et énergies

#### 3.2.1 Adaptation du code Python à l'étude d'un agglomérat de titania

Avant de procéder à l'exploitation des données, nous avons modifié le code afin de prendre en compte la nature de l'agglomérat utilisé. Cela a été fait en modifiant la valeur de la constante de Hamaker, en utilisant la formule suivante (Eq. 4) fournie dans le document de référence.

[1]

$$A_{131}(h) = A_{131,s} + (A_{131,0} - A_{131,s}) \cdot \left(1 + \left(\frac{h}{h_{cr}}\right)^{\frac{3}{2}}\right)^{-\frac{3}{2}} \quad (Eq. 4)$$

A l'origine, il s'agissait d'une constante que nous avons convertie en fonction dans le but de la généraliser pour mieux approcher les différents matériaux. Cette formule correspond à la fonction de Lifshitz-Hamaker (Eq. 4). Elle calcule la constante de Hamaker comme fonction de  $h$ . Elle prend en compte le retard du coefficient de Hamaker à grande distance, à travers le paramètre  $h_{cr}$ . Cette valeur est propre à la nature de l'agglomérat étudié. Les valeurs  $A_{131,s}$  et  $A_{131,0}$  sont aussi caractéristiques du matériau.

Cette formule a été intégrée au code (Fig. 7), ainsi que les paramètres relatifs à l'étude d'un agglomérat de nanoparticules de titania (Fig.6).

Système étudié	$A_{131,0}$ (J)	$A_{131,s}$ (J)	$h_{cr}$ (nm)
titania-eau-titania	$5,35 \times 10^{-20}$	$0,011 \times 10^{-20}$	23

Figure 6: paramètres de la fonction de Lifshitz-Hamaker pour des particules de titania (extrait du document de référence [1])

```

*****
# Hamaker constant
*****
def Hamaker(r, A_0, A_s, h_cr):
    A= A_s + (A_0 - A_s)*pow(1+pow(r/h_cr, 3/2), -2/3)
    return A

A_0 = 5.35e-20;          #! Non-retarded Hamaker constant (J)
A_s = 0.011e-20;       #! Static contribution of permanent dipoles (J)
h_cr = 23e-09;         #! Distance of retardation (m)

```

Figure 7: codage de la fonction de Liftshitz-Hamaker et paramètres relatifs au titania dans le code Python

### 3.2.2. Exploitation des résultats: modélisation des énergies et analyse des courbes

Dans un premier temps, nous avons analysé les fonctions majeures du programme fourni, destinées à calculer les énergies d'interaction entre deux agglomérats ainsi qu'à donner les caractéristiques de ces derniers. Au total, nous avons étudié 5 grandes fonctions du programme en profondeur et nous avons sélectionné les 3 plus pertinentes d'entre elles qui illustrent au mieux le rôle principal du code et qui s'accordent le plus à la globalité du projet.

Pour commencer, nous avons choisi la fonction appelée **total\_potential\_vol\_equivalent\_sph**. Le but de cette fonction est de retourner les valeurs des énergies potentielles (à savoir les valeurs d'origines répulsives et attractives) entre deux sphères de volume équivalent. Pour cela, la fonction calcule le volume total, le centre ainsi que la distance entre deux sphères  $i$  et  $j$  de manière à donner en sortie les valeurs énergies potentielles attractives et répulsives, grâce aux fonctions **Attractive\_potential** et **Repulsive\_potential** que nous allons détailler.

Le but de la fonction **Attractive\_potential** est de retourner la valeur d'énergie potentielle attractive entre deux sphères grâce à la formule de Van Der Waals (Eq. 3) et à la distance entre deux sphères et à leur rayon.

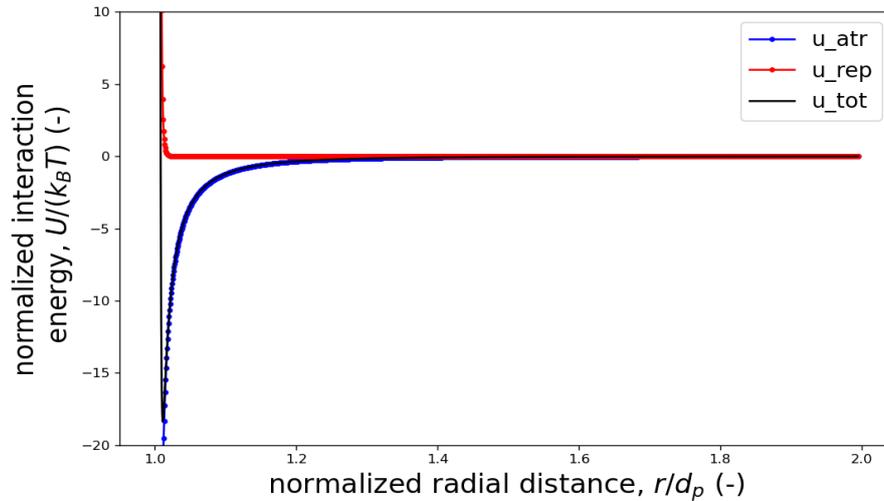
Enfin, l'objectif de la fonction **Repulsive\_potential** est de retourner la valeur de l'énergie potentielle répulsive entre deux sphères. La fonction prend en entrée le diamètre des sphères et la distance qui les sépare et utilise la formule de **Lennard-Jones : s\_LJ** (qui calcule le potentiel entre deux atomes) pour réaliser une intégration sur l'ensemble des atomes des deux sphères et ainsi trouver la valeur totale du potentiel répulsif. La formule de **Lennard-Jones [3]** utilisée grâce à une intégration sur tous les atomes est la suivante :

$$u_{mm}^{rep}(r) = \frac{A_{131}^6 \sigma_{LJ}}{2520r} \left\{ D_p^2 \left[ \frac{1}{2(r-D_p)^7} + \frac{1}{2(r+D_p)^7} + \frac{1}{r^7} \right] - \frac{D_p}{3} \left[ \frac{1}{(r-D_p)^6} + \frac{1}{(r+D_p)^6} \right] - \frac{1}{15} \left[ \frac{2}{r^5} - \frac{1}{(r-D_p)^5} - \frac{1}{(r+D_p)^5} \right] \right\} \quad (Eq. 5)$$

Après exécution du code, nous obtenons trois courbes caractérisant les énergies d'interaction. Ces trois courbes ont été réalisées pour des agglomérats de 100 particules, en utilisant les valeurs moyennes des paramètres fractals:  $D_f = 1.79$ ,  $k_f = 1.40$ . Nous avons effectué le programme avec 30 rotations, le nombre de rotations étant le nombre de directions utilisées par l'algorithme pour effectuer les calculs et pallier au problème de la forme anisotrope de l'agglomérat.

La première courbe (Fig.8) caractérise les interactions entre deux particules de titania. Ces énergies d'interaction sont normalisées par la grandeur  $k_B T$ , produit de la constante de Boltzmann et de la température du milieu. Cette grandeur représente l'énergie thermique du milieu dans lequel les particules sont en suspension. Le paramètre  $D_p$  correspond au diamètre d'une particule, qui vaut ici 30 nm, tandis que  $r$  est variable et correspond à la distance qui sépare les deux particules. Quand  $\frac{r}{D_p} = 1$ , la distance entre les deux sphères est minimale donc l'attraction est minimale et la répulsion maximale. En effet, les deux particules étant de même signe, à petite distance, elles auront tendance à vouloir s'éloigner l'une de l'autre. A l'inverse, quand la distance est maximale, pour une quantité  $\frac{r}{D_p} = 2$ , la répulsion est minimale par rapport

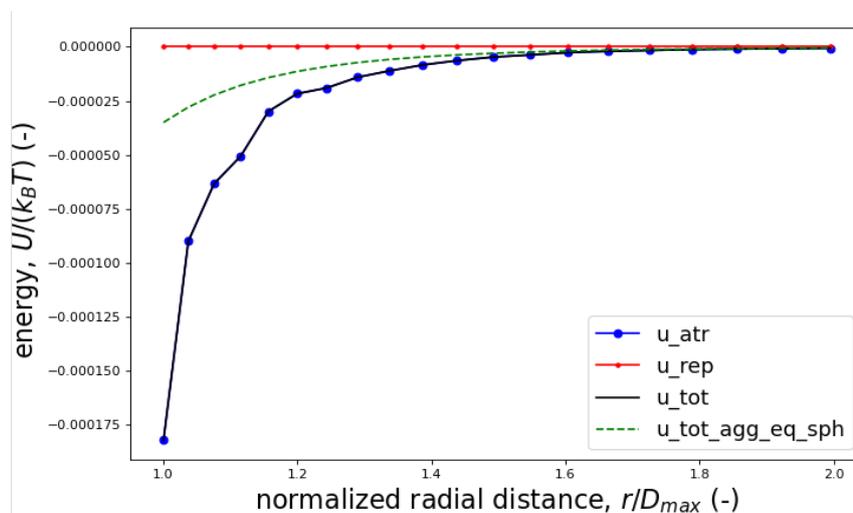
à l'attraction. Le potentiel d'attraction croit lorsque la distance entre les sphères diminue, tandis que le potentiel de répulsion aura le comportement inverse.



**Figure 8:** Graphe de l'énergie d'interaction en fonction de la distance entre deux particules de titania

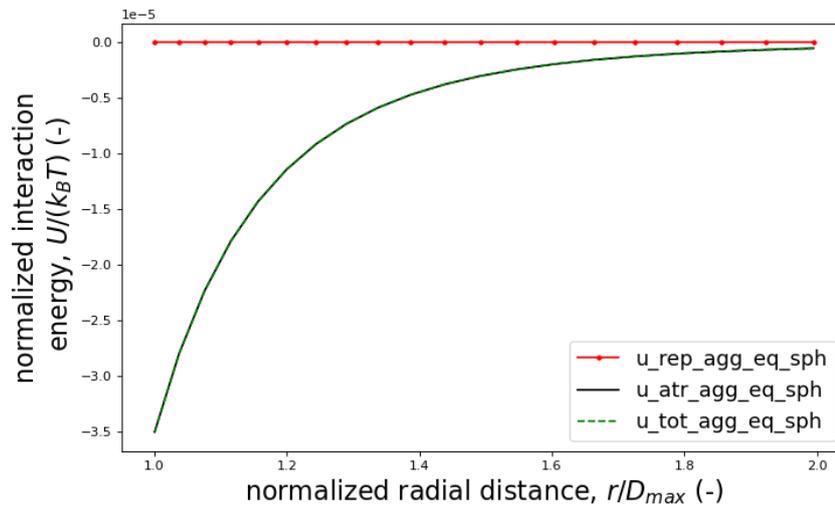
La deuxième courbe réalisée (Fig.9) modélise les énergies d'interaction entre deux agglomérats de nanoparticules de titania. Nous observons que l'énergie de répulsion est négligeable devant l'énergie d'attraction. Par ailleurs, plus la distance entre les agglomérats augmente, plus les forces attractives diminuent. Cela est cohérent avec le fonctionnement des interactions de VdW, qui sont non négligeables à courte distance.

Outre les énergies d'attraction, de répulsion et totale, nous disposons d'une quatrième courbe qui modélise les interactions totales dans le cas d'une représentation des agglomérats en volumes équivalents sphériques. L'équivalent sphérique modélise le volume global de l'agglomérat, c'est-à-dire la somme des volumes de toutes les particules qu'il contient, sous forme d'une sphère de volume équivalent. Ce modèle se rapproche des valeurs expérimentales lorsque la distance entre les agglomérats augmente. À grande échelle, il n'est pas valable car les interactions entre chaque particules ne peuvent pas être négligées.



**Figure 9:** Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de nanoparticules de titania

La troisième et dernière courbe (*Fig.10*) modélise l'énergie totale entre deux agglomérats suivant le modèle de l'équivalent sphérique. En effet, pour tracer cette courbe, nous utilisons exclusivement la fonction **total\_potential\_vol\_esquivalent\_sph**. Encore une fois, nous remarquons que l'énergie de répulsion peut être négligée devant l'énergie totale moyenne. De ce fait, nous pouvons considérer que les courbes de l'énergie d'attraction et d'énergie totale entre deux agglomérats, en suivant le modèle d'équivalent sphérique, sont confondues. Enfin, nous remarquons encore une fois que plus la distance augmente et plus l'énergie attractive diminue.



*Figure 10: Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de nanoparticules de titania, concentré sur l'énergie totale en équivalent sphérique*

### 3.3. Modélisation et variations des paramètres clés

#### 3.3.1. Modélisation

Après plusieurs simulations et études de cas sur différents agglomérats, nous avons cherché à modéliser les interactions de VdW d'un agglomérat à l'aide d'une formule théorique, pour vérifier la validité des courbes obtenues avec le code

Pour cela, nous avons utilisé la formule proposée par Naumann et Bunz (*Eq.6*). La dimension fractale de nos agglomérats étant de 1.8, nous pouvons donc utiliser cette formule valable en dimension fractale 2 en étant conscients que le résultat ne sera pas totalement identique à nos modélisations.

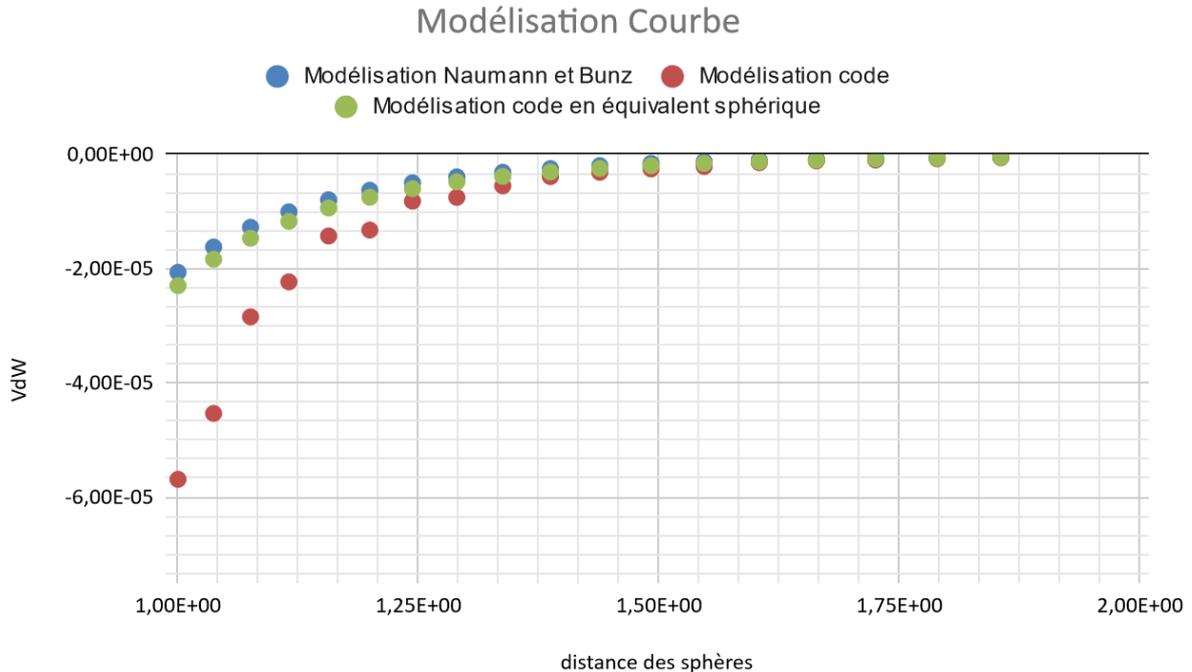
$$V_{vdW}(d) = -\frac{\varphi^2}{27} A_{131} \left( \frac{3x^2}{d^2} - \frac{16x^2}{4d^2 - x^2} + \frac{x^2}{d^2 - x^2} \right) \text{ (Eq. 6)}$$

Avec  $\varphi = \frac{Nx_p^3}{x^3}$

Nous avons extrait du fichier "Groupe25", ([voir annexe](#)) la distance  $d$  nécessaire à l'application de l'équation 6, également appelée  $r$  dans le fichier Python. Nous avons aussi extrait les valeurs des interactions de VdW, pour les rajouter sur le graphique (*Fig.11*) et

pouvoir comparer plus facilement les résultats obtenus avec le code et ceux obtenus avec la formule (Eq. 6)

Nous avons le même axe des abscisses que dans les graphiques obtenus avec le fichier Python (fig. 8 et 9), soit le rapport entre la distance  $d$  sur  $r_{max}$ .



*Figure 11 : Graphe de la modélisation de l'énergie de VdW en fonction de la distance entre les deux agglomérats, comparant les valeurs obtenues avec la formule de Naumann et Bunz aux valeurs obtenues avec le code.*

Nous pouvons constater que la formule fournie par Naumann et Bunz correspond bien au modèle attendu et confirme ainsi les résultats donnés par le programme Python.

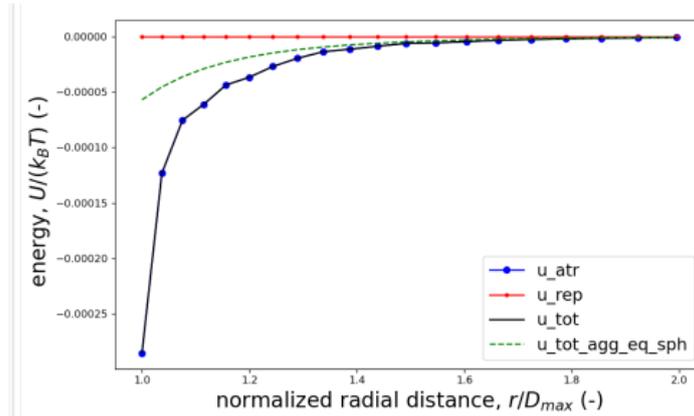
Par ailleurs, nous remarquons que nous nous rapprochons grandement de l'énergie totale en équivalent sphérique plus que de la force d'attraction dans les courtes distances. Cela n'est pas étonnant car, dans cette modélisation de Naumann et Bunz, la formule considère une distribution de masse sphérique malgré une dimension fractale égale à 2.

### 3.3.2 Variations des paramètres clés

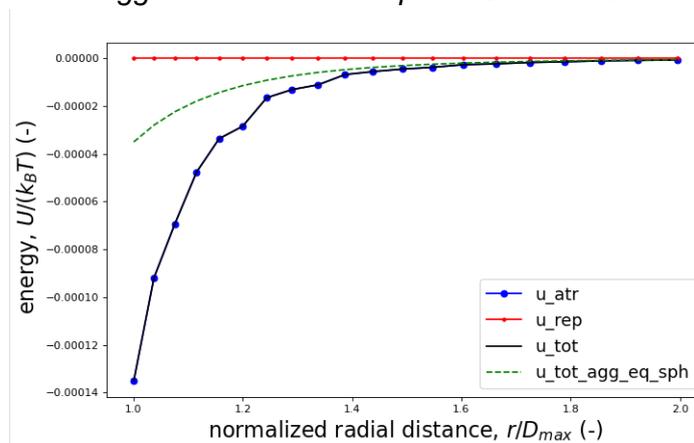
Dans l'optique de comparer nos différentes courbes, nous avons modifié les différents paramètres clés (Fig.4) du code Python que nous utilisons. Nous avons donc choisi de modifier le nombre de rotations, de particules et la valeur des paramètres fractals  $D_f$  et  $k_f$ .

#### 3.3.2.1 Variation du nombre de rotations

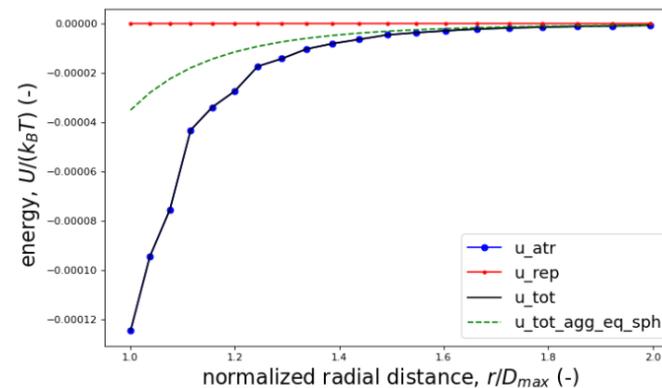
Dans un premier temps, nous faisons varier le nombre de rotations lors de l'exécution du programme tout en fixant le nombre de particules à 100.



**Figure 12 :** Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de titania pour 15 rotations.



**Figure 13:** Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de titania pour 50 rotations.

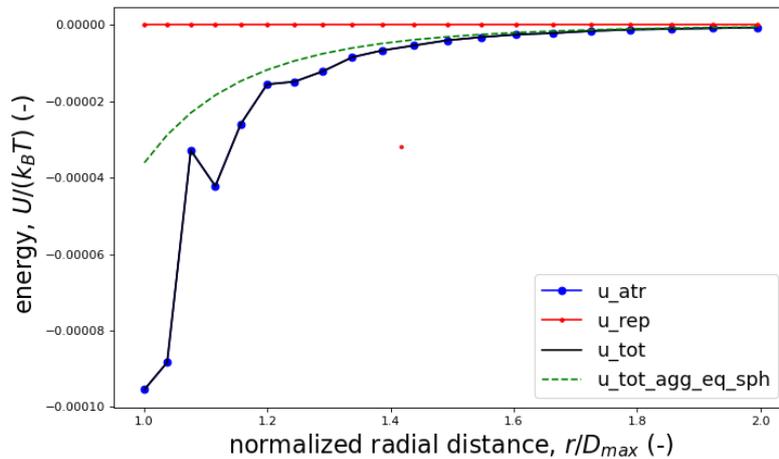


**Figure 14 :** Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de titania pour 100 rotations.

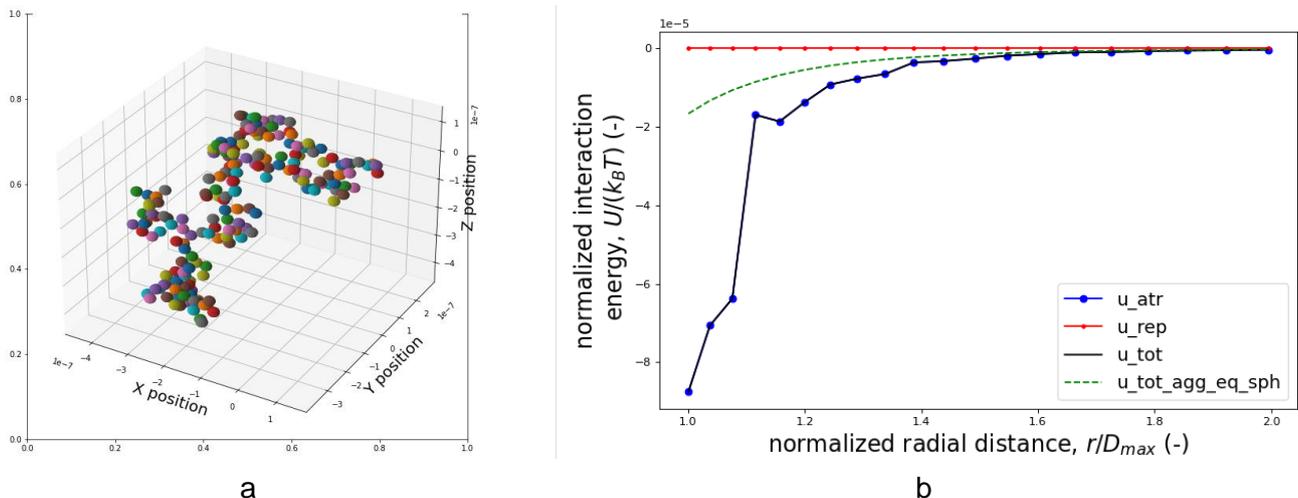
Ainsi, en augmentant le nombre de rotations, nous diminuons les oscillations présentes quand la distance radiale est très petite. Cependant, comme les graphes sont générés avec une partie aléatoire, selon les simulations les courbes sont plus ou moins lisses ; nous nous attendions à une courbe plus lisse pour 100 rotations. Nous constatons quand même que lorsque la distance augmente, les courbes sont sensiblement les mêmes, le nombre de rotations n'influence pas l'aspect de la courbe. En effet, à grande distance, nous nous rapprochons plus de l'équivalent sphérique, quel que soit le nombre de rotations.

### 3.3.2.2 Nombre de particules

Maintenant, nous fixons le nombre de rotations à 30, pour limiter le temps de compilation, et nous faisons varier le nombre de particules. De plus, en fixant le nombre de rotations à 30, nous pouvons comparer la courbe à la Fig. 9 .



**Figure 15:** Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de titania de 50 particules.



**Figure 16 a:** Modélisation d'un agglomérat de titania de 200 particules.

**Figure 16 b:** Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de titania de 200 particules chacun.

Nous constatons que la courbe des énergies d'attraction pour les agglomérats de 200 particules tend vers 0 de façon plus rapide que la même courbe pour des agglomérats de 50 particules. Notre hypothèse est que la distance entre chaque paire de particules étant accrue dans des gros agglomérats, cela diminue les forces de Van der Waals entre cette paire. Ainsi, la somme des forces de Van der Waals est plus petite pour les gros agglomérats. Cette hypothèse expliquerait aussi pourquoi l'énergie de l'équivalent sphérique de 50 particules est plus importante que l'énergie de l'équivalent sphérique de 200 particules.

### 3.3.2.3 Paramètre fractals ( $D_f$ et $k_f$ )

Jusqu'ici, les calculs étaient effectués avec des agglomérats de dimension fractale  $D_f = 1.79$  et  $k_f = 1.4$ . Les paramètres fractals déterminent la structure de l'agglomérat. Nous fixons le nombre de particules à 50 et le nombre de rotations à 30.

- Modification de  $D_f$  en fixant  $k_f$  à 1.4.

$D_f$  caractérise la compacité de l'agglomérat à l'échelle de l'agglomérat, c'est un facteur compris entre 1 et 3.

Autrement dit, quand  $D_f = 1$ , l'agglomérat a une structure linéaire tandis qu'une dimension fractale égale à 3 représente un agglomérat plus compact, proche d'une sphère.

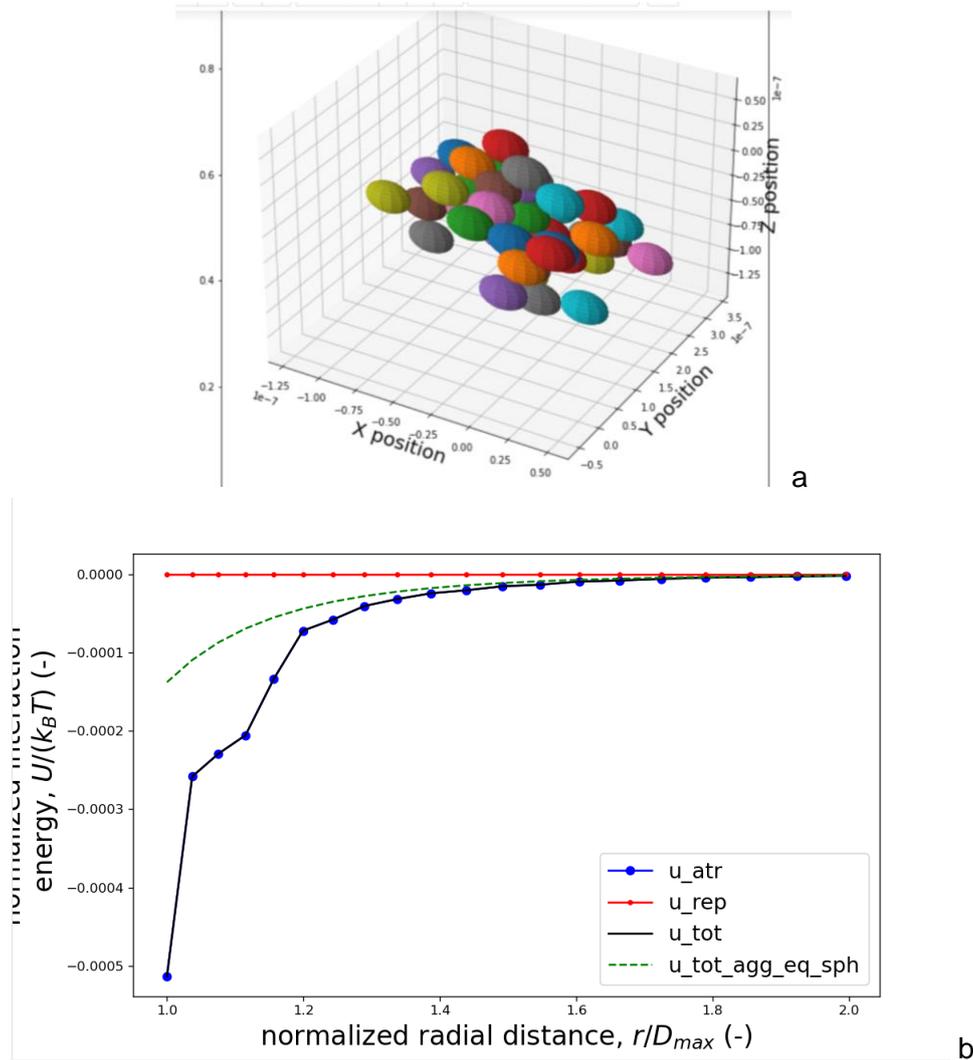
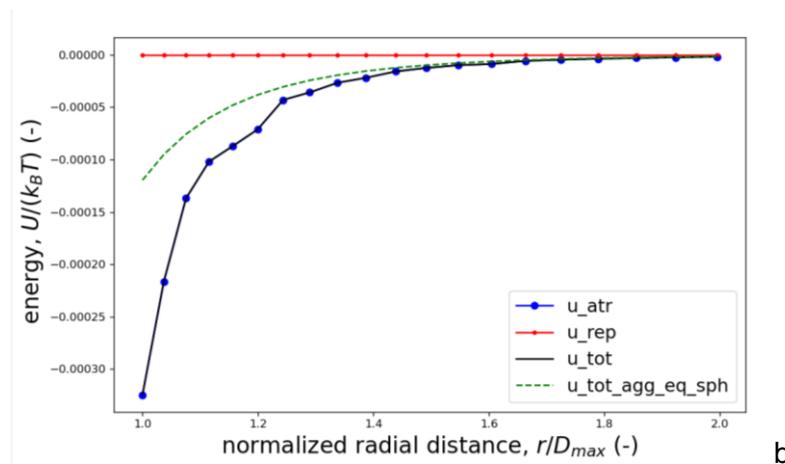
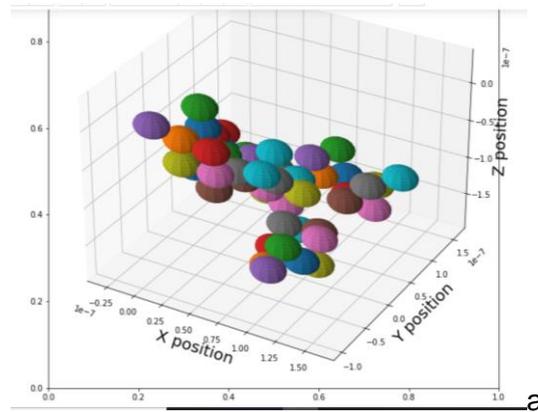


Figure 17a : Modélisation d'un agglomérat de titania de 50 particules avec  $D_f = 1.90$  et  $k_f = 1.4$

Figure 17 b : Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de titania de 50 particules chacun, avec  $D_f = 1.90$  et  $k_f = 1.4$



**Figure 18a :** Modélisation d'un agglomérat de titania de 50 particules avec  $D_f = 2$  et  $k_f = 1.4$

**Figure 18b :** Graphe de l'énergie d'interaction en fonction de la distance entre deux agglomérats de titania de 50 particules chacun, avec  $D_f = 2$  et  $k_f = 1.4$

Quand  $D_f$  augmente, les interactions se rapprochent de la courbe de l'équivalent sphérique, car nous nous rapprochons de la dimension 3 donc de la sphère. Nous trouvons bien la même courbe pour l'équivalent sphérique, quel que soit le  $D_f$ , ce qui est cohérent car nous avons le même nombre de particules donc le même volume équivalent sphérique.

Sur la représentation des agglomérats, nous constatons bien que l'agglomérat avec le plus grand  $D_f$  est plus éclaté dans l'espace, c'est cohérent avec la définition.

A partir de  $D_f = 2.2$ , il y a trop d'itérations, le programme boucle à l'infini, il est impossible de générer un fichier FracVAL et de pousser la modélisation plus loin.

- Modification de  $k_f$  en fixant  $D_f$  à 1,8

Le facteur  $k_f$  modifie la configuration des particules au sein de l'agglomérat, c'est un facteur compris entre 1 et 1,8.

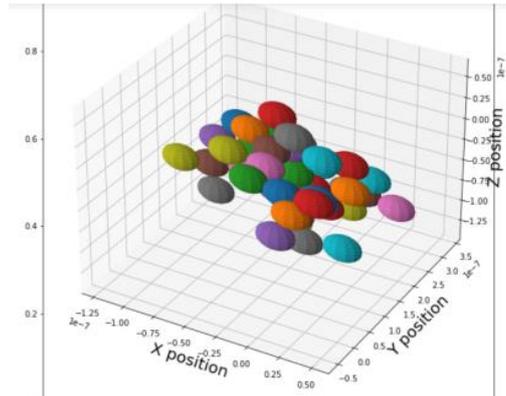


Figure 19 : Modélisation d'un agglomérat de titania de 50 particules avec  $D_f = 1.8$  et  $k_f = 1.6$

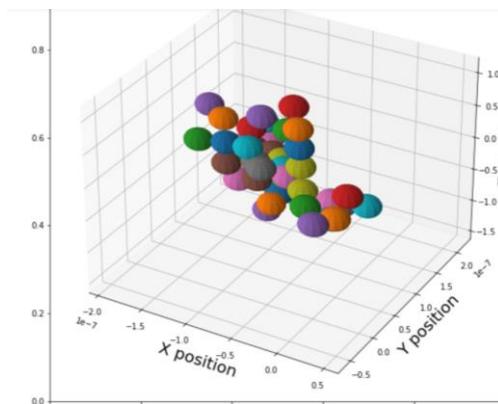


Figure 20 : Modélisation d'un agglomérat de titania de 50 particules avec  $D_f = 1.8$  et  $k_f = 1.4$

Augmenter  $k_f$  augmente la compacité de l'agglomérat à l'échelle locale, autrement dit les particules sont plus rapprochées mais la structure globale de l'agglomérat reste la même.

De même, à partir de  $k_f = 1.8$ , il est impossible de générer un fichier FracVAL car le programme ne peut pas gérer l'ensemble des itérations.

En somme, la modification de certains paramètres fait varier la disposition des particules au sein de l'agglomérat, comme  $k_f$  et  $D_f$ . Certains paramètres modifient les interactions entre les agglomérats, comme l'augmentation du nombre de particules dans un agglomérat, ou la modification des paramètres fractaux, qui font que nous nous rapprochons plus ou moins du modèle en volume sphérique.

Enfin, faire varier le nombre de rotations va juste augmenter la précision des calculs des interactions, donc le nombre d'itérations. L'augmentation du nombre de particules agit de la même façon sur le nombre d'itérations.

#### 4. CONCLUSION ET PERSPECTIVES

Ce projet de groupe est important dans l'apprentissage du métier d'ingénieur, il nous permet de nous intéresser à un sujet de physique plus approfondi et plus complexe. De plus, nous devons apprendre à travailler avec de nouvelles personnes en groupe sur des logiciels que nous découvrons. Après avoir été familiarisés avec le langage Pascal durant nos cours d'informatique, la compréhension du langage Python a été grandement facilitée. Ce projet nous a aussi appris à être autonome, efficace et à savoir poser des questions à notre professeur-référent.

Le projet nous a permis de découvrir le monde des nanoparticules, et de prendre conscience que les forces de VdW entre des agglomérats ne sont pas facilement modélisables. Pour cela, il est nécessaire de faire des approximations dans le modèle utilisé. Ici nous n'avons considéré que 2 agglomérats similaires, alors que pour être au plus proche de la réalité il faudrait considérer tout l'environnement d'un agglomérat. Nous avons aussi compris que ne considérer qu'un modèle en volume sphérique ne serait valable que pour des agglomérats séparés d'une grande distance. Le modèle de Naumann et Bunz est cohérent avec les courbes obtenues avec Python sur des courtes distances. En complément, nous pourrions étudier une autre nanoparticule qui nous était proposée, le silica, et comparer ses propriétés physiques à celles du titania.

Nous aimerions remercier notre professeur-référent Mr. Morán de nous avoir accompagné tout au long du semestre sur notre projet.

#### 5. BIBLIOGRAPHIE

[1] Document de référence : Frank Babick, Karin Schießl, Michael Stintz, "Van der Waals interaction between two fractal aggregates", *Masayoshi Fuji*, 220-225, 13 décembre 2010.

[2] F.J. Heiligtag & M. Niederberger, "The fascinating world of nanoparticle research" *Materials Today*, 16(7-8), 262-271, juillet-août 2013.

[3] Lorenzo Isella, Yannis Drossinos, "Langevin agglomeration of nanoparticles interacting via a central potential", *Physical review E82*, 011404, 2010.

#### 6. ANNEXE

Code en Python "Groupe25\_Interaction\_VDW" fourni par M. Morán

```
Z = 0.4e-09;           #! minimum separation (m)
rho_p = 1800          #! Particle bulk mass density (kg/m^3)
k_B = 1.38066e-23    #! Boltzmann constant in (J/K)
T = 300              #! Temperature (K)
k_BT = k_B*T         #! Average thermal energy
```

2. Open the aggregate

```
[3]: path = r"C:\Users\paul1\Documents\N_00000050_Agg_00000001.dat"
spheres = Read_spheres(path,1e-09,False)
spheres.head()
```

```
[3]:
```

	x	y	z	r	v
0	1.501187e-08	1.466980e-08	-2.440197e-08	1.500000e-08	1.413717e-23
1	3.216288e-08	3.995243e-09	-4.658070e-08	1.500000e-08	1.413717e-23
2	-2.074946e-09	-9.984802e-09	-2.396444e-08	1.500000e-08	1.413717e-23
3	5.333301e-08	-7.811653e-09	-2.890521e-08	1.500000e-08	1.413717e-23
4	3.084495e-08	-4.616974e-09	-7.528771e-08	1.500000e-08	1.413717e-23

```
[22]: Np = len(spheres)           #! Number of primary particles
Rp = np.mean(spheres["r"])      #! Primary particle radius
Dp = 2*Rp                       #! Primary particle diameter
```

```
[23]: x_cm, y_cm, z_cm = Center_mass(spheres)
```

### 1.1 Plot a characteristic interaction potential

Plot the characteristic interaction potential between two spheres.

```
[24]: radial_dist = np.logspace(0,0.3,1000)*Dp*1.000000001
u_atr = np.zeros_like(radial_dist, dtype=np.float)
u_rep = np.zeros_like(radial_dist, dtype=np.float)
u_tot = np.zeros_like(radial_dist, dtype=np.float)
```

```
[25]: i=0
for r in radial_dist:
    A = Hamaker(r, A_0, A_s, h_cr);           #! Hamaker constant (J)
    u_atr[i] = Attractive_potential(Dp,r,A)
    u_rep[i] = Repulsive_potential(Dp,r,A,s_LJ)
    u_tot[i] = u_atr[i]+u_rep[i]
    i=i+1
```

```
[26]: fig = plt.figure(num=None, figsize=(10, 6), dpi=80, facecolor='w',
    -edgecolor='k')
fig.set_tight_layout(False)
plt.plot(radial_dist/Dp, u_atr/k_BT, "-b",label="u_atr")
plt.plot(radial_dist/Dp, u_rep/k_BT, "-r",label="u_rep")
```

```
plt.plot(radial_dist/Dp, u_tot/k_BT, "-k",label="u_tot")
plt.xlabel('normalized radial distance, $r/d_p$ (-)', fontsize=20)
plt.ylabel('normalized interaction\ n energy, $U/(k_BT)$ (-)', fontsize=20)
plt.ylim([-20,10])
#plt.xlim([0.9,2])
plt.legend(fontsize=16, loc=0), plt.show()
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

[26]: (<matplotlib.legend.Legend at 0x2a13b209040>, None)

## 1.2 Calculate the interaction potentials

```
[27]: spheres_copy = spheres.copy()
spheres_copy.head()
```

```
[27]:
```

	x	y	z	r	v
0	1.501187e-08	1.466980e-08	-2.440197e-08	1.500000e-08	1.413717e-23
1	3.216288e-08	3.995243e-09	-4.658070e-08	1.500000e-08	1.413717e-23
2	-2.074946e-09	-9.984802e-09	-2.396444e-08	1.500000e-08	1.413717e-23
3	5.333301e-08	-7.811653e-09	-2.890521e-08	1.500000e-08	1.413717e-23
4	3.084495e-08	-4.616974e-09	-7.528771e-08	1.500000e-08	1.413717e-23

```
[28]: Rmax = Rmax_spheres(spheres)
Dmax = 2*Rmax

Dv = np.power(np.sum(spheres["v"])*6.0/np.pi,1.0/3.0)
```

```
[29]: distance = np.logspace(0,0.3,20)*Dmax
```

```
[30]: v_i = np.sum(spheres["v"])
dv_i = np.power(6.0*v_i/np.pi,1.0/3.0)
dv_i/Dmax
```

[30]: 0.2545004206021001

```
[31]: u_atr_agg = np.zeros_like(distance, dtype=np.float)
u_rep_agg = np.zeros_like(distance, dtype=np.float)
u_tot_agg = np.zeros_like(distance, dtype=np.float)

u_atr_min_agg = np.zeros_like(distance, dtype=np.float)

u_atr_agg_eq_sph = np.zeros_like(distance, dtype=np.float)
u_rep_agg_eq_sph = np.zeros_like(distance, dtype=np.float)
u_tot_agg_eq_sph = np.zeros_like(distance, dtype=np.float)
```

```
[32]: rotations = 30

[33]: i=0
for r in tqdm(distance):
    u_atr_i_avg = 0.0
    u_rep_i_avg = 0.0
    u_atr_i_min = 1e+10
    for j in range(rotations):
        point_sph = rand_point_sphere()
        spheres_copy["x"] = spheres["x"] + r*point_sph[0]
        spheres_copy["y"] = spheres["y"] + r*point_sph[1]
        spheres_copy["z"] = spheres["z"] + r*point_sph[2]
        u_atr_i, u_rep_i = total_potentials_agg_ij(spheres,spheres_copy,A,s_LJ)
        if (u_atr_i < u_atr_i_min):
            u_atr_i_min = u_atr_i
            u_atr_i_avg = u_atr_i_avg + u_atr_i
            u_rep_i_avg = u_rep_i_avg + u_rep_i
    u_atr_i_avg = u_atr_i_avg/rotations
    u_rep_i_avg = u_rep_i_avg/rotations
    u_atr_min_agg[i] = u_atr_i_min
    u_atr_agg[i] = u_atr_i_avg
    u_rep_agg[i] = u_rep_i_avg
    u_tot_agg[i] = u_atr_i_avg+u_rep_i_avg
    u_atr_agg_eq_sph[i],u_rep_agg_eq_sph[i] = u
    total_potential_vol_equivalent_sph(spheres,spheres_copy,A,s_LJ)
    i = i+1

100% | 20/20 [01:22<00:00, 4.12s/it]

[34]: u_tot_agg_eq_sph = u_atr_agg_eq_sph+u_rep_agg_eq_sph

[35]: fig = plt.figure(num=None, figsize=(10, 6), dpi=80, facecolor='w',
    edgecolor='k')
fig.set_tight_layout(False)
plt.plot(distance/Dmax, u_atr_agg/k_BT, "o-b",label="u_atr")
plt.plot(distance/Dmax, u_rep_agg/k_BT, ".-r",label="u_rep")
plt.plot(distance/Dmax, u_tot_agg/k_BT, "-k",label="u_tot")
plt.plot(distance/Dmax, u_tot_agg_eq_sph/k_BT, "--g",label="u_tot_agg_eq_sph")
plt.xlabel('normalized radial distance, $r/D_{max}$ (-)', fontsize=20)
plt.ylabel('normalized interaction energy, $U/(k_{BT})$ (-)', fontsize=20)
plt.ylim([-20,10])
plt.legend(fontsize=16, loc=0), plt.show()

<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>

[35]: (<matplotlib.legend.Legend at 0x2a13b20fdf0>, None)
```

```
[36]: fig = plt.figure(num=None, figsize=(10, 6), dpi=80, facecolor='w',
    edgecolor='k')
    fig.set_tight_layout(False)
    plt.plot(distance/Dmax, u_rep_agg_eq_sph/k_BT, "-r",label="u_rep_agg_eq_sph")
    plt.plot(distance/Dmax, u_atr_agg_eq_sph/k_BT, "-k",label="u_atr_agg_eq_sph")
    plt.plot(distance/Dmax, u_tot_agg_eq_sph/k_BT, "--g",label="u_tot_agg_eq_sph")
    plt.xlabel('normalized radial distance, $r/D_{max}$ (-)', fontsize=20)
    plt.ylabel('normalized interaction energy, $U/(k_BT)$ (-)', fontsize=20)
    #plt.ylim([-20,10])
    plt.legend(fontsize=16, loc=0), plt.show()
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

[36]: (<matplotlib.legend.Legend at 0x2a13bff7160>, None)

### 1.3 Save results

```
[39]: all_data = np.column_stack((distance, u_atr_agg/k_BT, u_atr_agg_eq_sph/k_BT))
    name_exp = "pourjustine.dat"
    np.savetxt(name_exp, all_data, delimiter=",")
```

[ ]:

[ ]: