Compte-rendu Analyse de données

CHOPIN Antonio

Table des matières

1 Introduction

L'objectif est d'étudier la densité atomique dans un système constitué d'atomes de silicium et d'oxygène à l'état liquide et solide.

Pour ce faire, on utilise l'outil mathématique **densité radiale** ou **corrélation de paires** $g_{\alpha\beta}(r)$ où α et β sont des espèces chimiques d'atomes et r une distance. Plus précisément, si on se concentre sur une particule de référence, cette fonction décrit la répartition moyenne dans l'espace des autres particules plus on s'en éloigne.

Il faut donc implémenter un code pour calculer cette fonction et la visualiser.

Ce document se décompose en trois parties : une première sur la compréhension, la création de l'algorithme permettant de calculer la fonction de densité et l'implémentation du code correspondant. Une seconde sur les résultats obtenus en utilisant des données issues de simulations préalablement effectuées. Enfin, une troisième sur l'interprétation des comportements observés pour les différents cas étudiés.

En plus de ce document, on fournit le dossier **Data-Analysis** contenant :

* **Les codes** et **fichiers de données** ayant permis d'obtenir les résultats présentés
* Le dossier **Résultats** contenant :
  — Le dossier **300K**, avec les résultats obtenus pour la silice à T = 300K
  — Le dossier **3500K**, avec les résultats obtenus pour la silice à T = 3500 K

Le langage utilisé pour les implémentations est le **c**
Les codes sont à exécuter avec **gcc <nomfichier> -lm**
Les figures obtenues sont tracées avec **gnuplot**

2 La fonction de distribution radiale

On présente ici les aspects théoriques et conceptuels nécessaires à la mise en place du code permettant de calculer la fonction de densité radiale puis les techniques mises en oeuvres pour son implémentation.

2.1 Concept

La fonction g(r) peut être pensée comme suit : Si on veut étudier la corrélation des paires d'atomes d'espèces α/β, alors pour un atome de type α de référence, g(r) retourne la "densité de présence" des atomes β à la distance r.

D'un point de vue algorithmique, si on a un système constitué d'atomes identiques, il faut donc parcourir toutes les paires de particules possibles afin de calculer les distances inter-atomiques. On peut alors créer un histogramme des résultats, autrement dit, visualiser la répartition spatiale des particules les unes par rapport aux autres par intervalle $[r; r + dr]$ où dr est un pas d'espace choisi pour discrétiser l'espace. Ce premier histogramme est représenté par une foncion h(r). On le normalise ensuite par rapport à un gaz parfait où les histogrammes des particules sont non corrélés. On obtient ainsi g(r).

Si on a affaire à un système avec différents types d'atomes, il faut faire attention à étudier les paires correctes pour créer un histogramme représentant effectivement la distribution que l'on veut étudier.

2.2 Algorithme général

On calcule donc d'abord **l'histogramme brut** h(r) puis g(r) est obtenu par **normalisation de h(r)**. h(r) et g(r) ont le même domaine de définition.

L'histogramme doit couvrir une certaine gamme de distances variant de 0 à une certaine valeur notée r_{max}. On choisit une distance maximale égale à la moitié de la taille de la boîte de côté L. On définit ainsi $r_{max} = \frac{L}{2}$. On divise ensuite cette valeur par un nombre $nk \in \mathbb{N}$ et on obtient le pas spatial dr. On a donc la relation :

$$\boxed{nk \times dr = r_{max}} \tag{1}$$

h et g permettront donc de couvrir la gamme de distance bornée par 0 et $\frac{L}{2}$.

Dans le cas étudié, le système est composé de 1008 atomes de deux espèces différentes : 672 d'oxygène et 336 de silicium. On veut obtenir g(r) pour trois cas :

— Les paires Si - O
— Les paires Si - Si
— Les paires O - O

On présente ici un pseudo-code (Algorithme 1) symbolisant les étapes principales du calcul de g(r) pour un système où sont présentes deux espèces d'atomes α et β. L'algorithme est adapté au cas à l'on veut étudier les paires α - β

Algorithme 1 : Algorithme pour calculer g(r)

/* N_α : nombre d'atomes de type 1 */
/* N_β : nombre d'atomes de type 2 */
/* $N = N_\alpha + N_\beta$: nombre total d'atomes */
/* L : la taille de la boîte de simulation */
/* h et g sont des tableaux */
Data : $N_\alpha, N_\beta, N, L, nk, h(nk)$
Result : $g(nk)$

/* Le facteur de normalisation */
$factor = \frac{L^3}{4\pi N_{\alpha\beta \times nb\_config}}$;

/* Parcours des paires α-β */
for $i = 1$ **to** N_α **do**
  for $j = N_\alpha$ **to** N **do**
    /* Différences de coordonnées */
    $dx = x_i - x_j$;
    $dy = y_i - y_j$;
    $dz = z_i - z_j$;

    /* Conditions aux limites périodiques */
    $dx = dx - Arrondi(\frac{dx}{L}) * L$;
    $dy = dy - Arrondi(\frac{dy}{L}) * L$;
    $dz = dz - Arrondi(\frac{dz}{L}) * L$;

    /* Calcul de la distance entre les atomes i et j */
    $r = \sqrt{dx^2 + dy^2 + dz^2}$;

    /* Calcul de l'indice k */
    $k = ceil(\frac{r}{dr})$;

    /* Remplissage de h[k] */
    if $k \leq nk$ **then**
      $h[k] = h[k] + 1$;

/* Normalisation de h(r) et obtention de g(r) */
for $i = 1$ **to** nk **do**
  $distance = dr * i$;
  $Vdr = (dr * distance^2)$;

  /* Remplissage de g[i] */
  $g[i] = \frac{h[i] * factor}{Vdr}$

Les histogrammes h(nk) et g(nk) résultent de **la moyenne de plusieurs configurations à différents instants** *nb_ config*.

Dans le cas où on veut étudier les paires d'atomes d'une même espèce, il sera nécessaire de changer les indices des boucles sur i et j calculant h et d'adapter le facteur de normalisation *factor*.

Comme indiqué, ce facteur est calculé selon l'expression :

$$\boxed{factor = \frac{L^3}{4\pi N_{\alpha\beta \times nb\_config}}} \tag{2}$$

Où α et β représentent des espèces atomiques et *nb_ config* est le nombre de configuration à partir desquelles est moyennée la fonction g(r).

$N_{\alpha\beta}$ est à adapter selon les cas :

$$\boxed{N_{\alpha\beta} = \begin{cases} N_\alpha(N_\alpha - 1) & \alpha = \beta \\ N_\alpha N_\beta & \alpha \neq \beta \end{cases}} \tag{3}$$

2.3 Implémentation

On implémente un code permettant de calculer la densité radiale g(r) en se basant sur l'algorithme 1. Les codes dans boucles principales pour les trois cas Si - O, Si - Si et O - O sont donnés en annexes (section A).

x, y et z sont des tableaux contenant les coordonnées des particules. Leur taille est de $1008 \times nb\_config$ (101 ici).

- Le cas Si-O est le cas traité par l'algorithme 1.

- Dans le cas de Si-Si, les indices de boucle sont modifiés de sorte à ne lire que les coordonnées correspondantes à des atomes de silicium. Les atomes d'oxygène ne sont pas pris en compte.

- Dans le cas O-O, ce sont les atomes de silicium qui ne sont pas pris en compte.

Pour une meilleure compréhension des codes, le lecteur est invité à se référer aux fichiers .c fournis en plus du présent document.

3 Résultats obtenus

L'algortihme de calcul de g(r) ayant été implémenté, on peut désormais l'appliquer aux deux cas étudié : de la Silice à l'état solide (T = 300 K) et à l'état liquide (T = 3500 K).

On présente ici les résultats obtenus pour la fonction de distribution spatiale g(r) grâce à la technique décrite dans la section précédente.

Les codes sont exécutés avec **gcc <nom fichier> - lm**

3.1 Cas Silice solide

On étudie les paires Si-O, O-O et Si-Si afin d'avoir une idée de la répartition des espèces atomiques dans le solide. On exécute l'algorithme adapté pour chaque cas et on trace les résultats sur un même graphe.

On obtient ainsi la figure 1

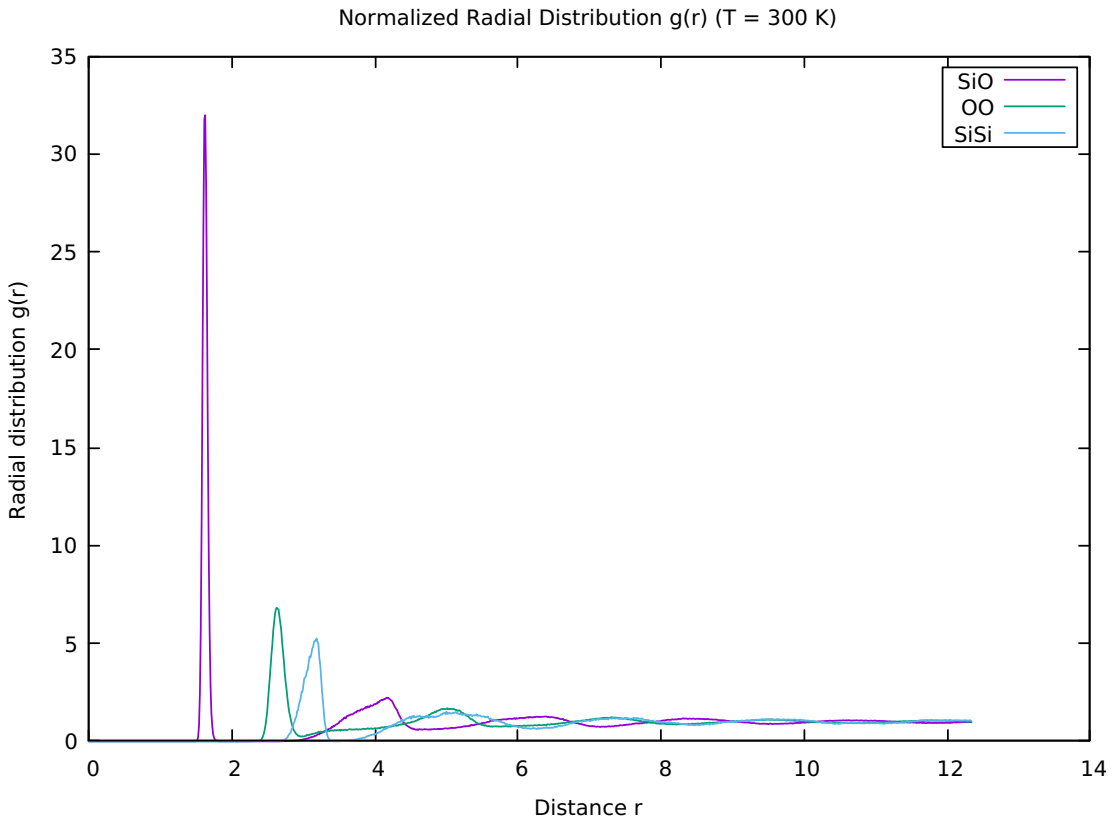

FIGURE 1 – Fonction radiale pour les différentes paires d'atomes (Cas T = 300 K)

La comparaison avec les figures présentées en cours permet d'affirmer que la normalisation a été correctement effectuée. De plus, les trois cas convergent vers 1 plus la distance r augmente.

- Dans le **cas Si-O**, les atomes semblent être assez présents pour $r \approx 1.7$ où $g(r) \approx 31$. Par la suite, la densité diminue brusquement puis augmente à nouveau légèrement pour $r \approx 4$ mais de façon beaucoup moins prononcée ($g(r) \approx 2$). Pour $r \geq 4.1$, g(r) converge vers 1.

- Dans le **cas O-O**, il n'y a qu'un seul pic de densité pour $r \approx 2.3$ où $g(r) \approx 7$ puis la densité diminue brusquement et tend vers 1 plus r augmente.

- Dans le cas **Si-Si**, on a un comportement similaire à O-O mais décalé vers la droite :

un pic à $r \approx 3$ où $g(r) \approx 6$, puis une chute de densité suivi de la convergence vers 1 plus r augmente.

3.2 Cas Silice liquide

On étudie les paires Si-O, O-O et Si-Si afin d'avoir une idée de la répartition des espèces atomiques dans le liquide. On exécute l'algorithme adapté pour chaque cas et on trace les résultats sur un même graphe.

On obtient ainsi la figure 2

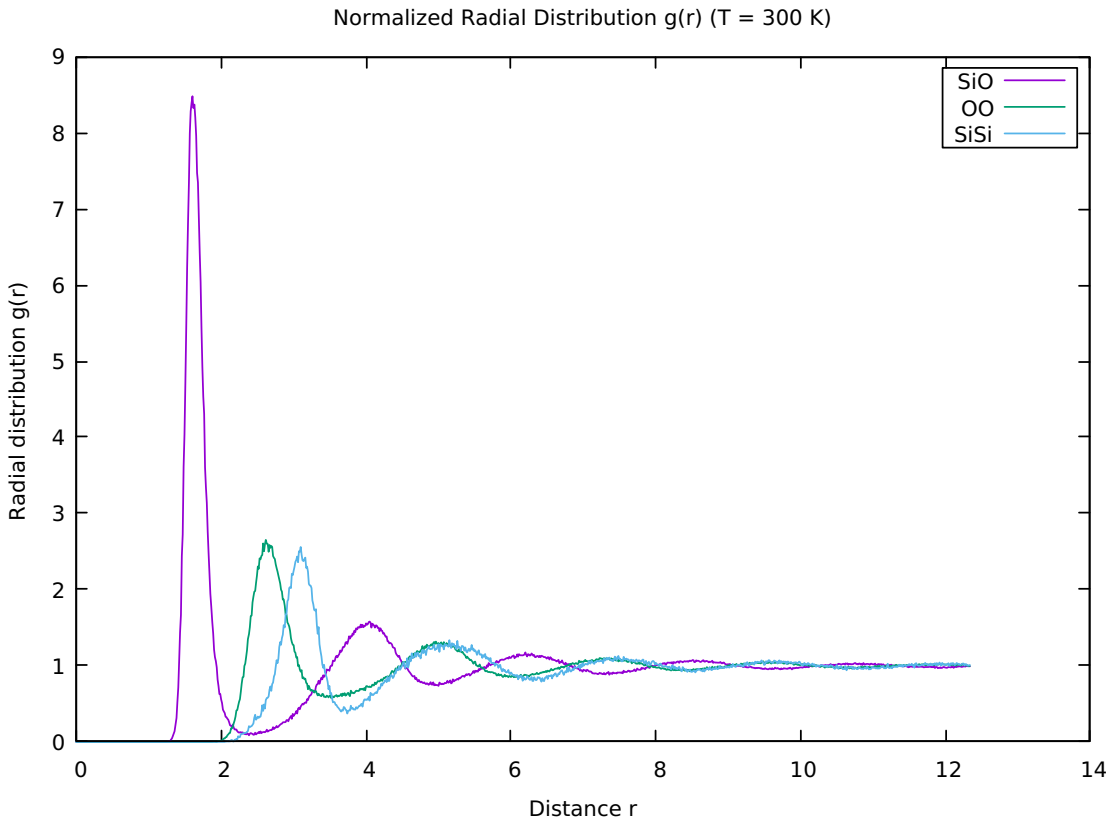

FIGURE 2 – Fonction radiale pour les différentes paires d'atomes (Cas T = 3500 K)

Les trois cas convergent vers 1 plus la distance r augmente. On peut donc dire que la normalisation a été correctement effectuée.

- Dans le **cas Si-O**, les atomes semblent assez présents pour $r \approx 1.8$ où $g(r) \approx 8.5$. Par la suite, la densité diminue brusquement puis augmente à nouveau légèrement pour $r \approx 4$ mais de façon beaucoup moins prononcée ($g(r) \approx 1.6$). Pour $r \geq 4.5$, g(r) converge vers 1.

- Dans le **cas O-O**, il n'y a qu'un seul pic de densité pour $r \approx 2.3$ où $g(r) \approx 2.7$ puis la densité diminue brusquement et tend vers 1 plus r augmente.

- Dans le cas **Si-Si**, on a un comportement similaire à O-O mais décalé vers la droite :

un pic à $r \approx 3$ où $g(r) \approx 2.6$, puis une chute de densité suivi de la convergence vers 1 plus r augmente.

Les comportements généraux sont donc assez similaires au cas solide.

4 Conclusion

L'algorithme 1 a bien permis de calculer puis de visualiser la fonction de densité radiale pour les différents cas étudiés.

De façon générale, on peut dire que les répartitions atomiques entre les différentes paires possibles dans la Silice liquide et solide sont similaires. Les valeurs atteintes dans le cas solide sont plus élevées cependant.

Le pic de la liaison Si-O laisse penser que la liaison entre ces deux espèces est plus courte que pour O-O et Si-Si. De plus, la valeur atteinte, bien supérieure aux autres types de paires, traduit peut-être la tendance des atomes de Si et O à se lier entre eux plus facilement que ne le font deux Si ou deux O.

Dans les deux cas solides et liquides, les paires Si-Si et O-O ont des comportement similaires mais décalés. Il semble que deux Silicium qui se lient entre eux ont une longueur de liaison plus élevée que celle de deux Oxygène.

A Boucle principale du programme

A.1 Cas des paires Si - O

```
// Calcul de l'histogramme h[nk]
    double dx, dy, dz, r;

  // Boucle sur le nombre de configurations
  for (num_it = 0; num_it < nb_config; num_it++)
  {
    // Boucle sur le premier type d'atomes
    for(i = num_it*N; i < num_it*N + N_O; i++)
    {
      // Boucle sur le second type d'atomes
      for(j = num_it*N + N_O; j < num_it*N + N; j++)
      {
        // Calcul différences des coordonnées
        dx = x[i] - x[j];
        dy = y[i] - y[j];
        dz = z[i] - z[j];

        // PBC
        dx = dx - round(dx/L) * L;
        dy = dy - round(dy/L) * L;
        dz = dz - round(dz/L) * L;

        // Calcul distance entre i et j
        r = sqrt(dx*dx + dy*dy + dz*dz);

        // Calcul de l'indice k
        k = ceil((r/dr));

        // Remplissage de h[k]
        if(k <= nk)
        {
          h[k] += 1;
        }
      }
    }
  }

  // Normalisation de h[nk]
  double Vdr;

  for(i = 1; i < nk; i++)
  {
    r = dr*i;
    Vdr = (dr*r*r);
    // Remplissage de g[i]
    g[i] = h[i]*factor_norm / Vdr;
  }
```

A.2 Cas des paires Si - Si

```
// Calcul de l'histogramme h[nk]
    double dx, dy, dz, r;

  // Boucle sur le nombre de configurations
  for (num_it = 0; num_it < nb_config; num_it++)
```

```
  {
    for(i = num_it*N + N_O; i < num_it*N + N; i++)
    {
      for(j = num_it*N + N_O; j < num_it*N + N; j++)
      {
        // Calcul différences des coordonnées
        dx = x[i] - x[j];
        dy = y[i] - y[j];
        dz = z[i] - z[j];

        // PBC
        dx = dx - round(dx/L) * L;
        dy = dy - round(dy/L) * L;
        dz = dz - round(dz/L) * L;

        // Calcul distance entre i et j
        r = sqrt(dx*dx + dy*dy + dz*dz);

        // Calcul de l'indice k
        k = ceil((r/dr));

        // Remplissage de h[k]
        if(k <= nk)
        {
          h[k] += 1;
        }
      }
    }
  }

  // Normalisation de h[nk]
  double Vdr;

  for(i = 1; i < nk; i++)
  {
    r = dr*i;
    Vdr = (dr*r*r);
    // Remplissage de g[i]
    g[i] = h[i]*factor_norm / Vdr;
  }
```

A.3 Cas des paires O - O

```
// Calcul de l'histogramme h[nk]
    double dx, dy, dz, r;

  // Boucle sur le nombre de configurations
  for (num_it = 0; num_it < nb_config; num_it++)
  {
    for(i = num_it*N; i < num_it*N + N_O; i++)
    {
      for(j = num_it*N; j < num_it*N + N_O; j++)
      {
        // Calcul différences des coordonnées
        dx = x[i] - x[j];
        dy = y[i] - y[j];
        dz = z[i] - z[j];

```

```
        // PBC
        dx = dx - round(dx/L) * L;
        dy = dy - round(dy/L) * L;
        dz = dz - round(dz/L) * L;

        // Calcul distance entre i et j
        r = sqrt(dx*dx + dy*dy + dz*dz);

        // Calcul de l'indice k
        k = ceil((r/dr));

        // Remplissage de h[k]
        if(k <= nk)
        {
          h[k] += 1;
        }
      }
    }
  }

  // Normalisation de h[nk]
  double Vdr;

  for(i = 1; i < nk; i++)
  {
    r = dr*i;
    Vdr = (dr*r*r);
    // Remplissage de g[i]
    g[i] = h[i]*factor_norm / Vdr;
  }
```