

Projet Minimisation

Kevin MOSCA

Ragousandirane RADJASANDIRANE

M1BI

1 Introduction

En dynamique des macromolécules on associe aux macromolécules une énergie potentielle. Cette énergie potentielle est calculée à l'aide de la formule présente sur la Figure 1 ($E_{pot} = E_{liaisons} + E_{angles} + E_{diedre} + E_{vdw} + E_{elec}$), et nous cherchons à minimiser cette valeur d'énergie potentielle. Pour ce faire il existe différents algorithmes d'optimisations tels que les techniques de Steepest Descent ou de Gradient Conjugué, qui réalisent une minimisation de l'énergie. Ici nous allons nous intéresser à la réalisation d'un algorithme de minimisation de type Steepest Descent sur une molécule d'eau en prenant en entrée un fichier .pdb comportant les données de coordonnées d'une molécule d'eau et en enregistrant les coordonnées après minimisation dans un autre fichier.pdb. Nous avons aussi tenté d'implémenter une minimisation de deux molécules d'eau.

Récapitulatif

$$\begin{aligned}
 V_{potential} = & \sum_{\text{toutes les liaisons}} \frac{k}{2} (l - l_0)^2 + \sum_{\text{tous les angles de valence}} \frac{k_\theta}{2} (\theta - \theta_0)^2 \\
 & + \sum_{\text{tous les dièdres}} \sum_{m=1}^M \frac{V_m}{2} [1 + \cos(m\omega - \gamma)] \\
 & + \sum_{i=1}^N \sum_{j=i}^N 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \\
 & + \sum_{i=1}^N \sum_{j=i}^N \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} + V_{user}
 \end{aligned}$$

termes liés

termes non liés

constantes du champ de forces

variables : géométrie de la molécule

PF 02/2021

43

FIGURE 1 – Formule de l'énergie potentielle

2 Matériels & Méthodes

2.1 Steepest decent

Nous utiliserons l'équation suivante pour notre algorithme de steepest decent :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{g}_k \quad (1)$$

On cherche à obtenir les positions du pas suivant \mathbf{x}_{k+1} en fonction des positions du pas actuel \mathbf{x}_k , du gradient des positions actuelles \mathbf{g}_k et d'un facteur λ qui représente le pas (dans notre cas, $\lambda = 10^{-4}$). On réalise le calcul des positions du pas suivant tant que le GRMS du gradient \mathbf{g}_k est supérieur à un certain seuil (dans notre cas, seuil = 0.01).

2.2 Gradient

Voici la représentation du gradient avec un exemple de calcul du premier terme :

$$\vec{g} = \begin{pmatrix} \frac{\partial E}{\partial x_{OH2}} \\ \frac{\partial E}{\partial y_{OH2}} \\ \frac{\partial E}{\partial z_{OH2}} \\ \frac{\partial E}{\partial x_{H1}} \\ \frac{\partial E}{\partial y_{H1}} \\ \frac{\partial E}{\partial z_{H1}} \\ \frac{\partial E}{\partial x_{H2}} \\ \frac{\partial E}{\partial y_{H2}} \\ \frac{\partial E}{\partial z_{H2}} \end{pmatrix} \quad avec \quad \frac{\partial E}{\partial x_{OH2}} = \frac{E(x_{OH2} - \delta, y_{OH2}, z_{OH2}) - E(x_{OH2} + \delta, y_{OH2}, z_{OH2})}{2\delta} \quad (2)$$

Chaque terme du gradient est une dérivation (numérique) partielle de l'énergie potentielle selon une coordonnée. On calcule donc la différence des énergies potentielles avec des variations infimes de δ sur chacune des coordonnées (dans notre cas, $\delta = \lambda/10$).

2.3 Coordonnées du système

Les coordonnées \mathbf{x}_k du système sont obtenus via un fichier pdb donné en entrée du script (argument 1) et sont stockées sous forme de liste de dictionnaire, avec un dictionnaire par atome.

2.4 Algorithme

Notre script utilise les modules python :

- **math** pour les calculs d'angles, racine...
- **copy** pour effectuer des copies conformes de liste de dictionnaire avec la fonction **deepcopy()**
- **matplotlib** pour réaliser des graphiques
- **sys** pour récupérer des arguments de console

Voici une présentation des différentes fonctions présentes dans le script :

2.4.1 read_pdb et write_pdb

Lit l'argument 1 (argv[1] qui correspond au fichier pdb d'entrée qui va permettre d'obtenir des positions xyz d'un system pour chaque atomes.

write_pdb permet d'écrire un fichier pdb avec une liste de dictionnaire des positions en entrée. Le nom du fichier pdb en sortie sera le nom donné en argument 2 de la ligne de commande (argv[2])

2.4.2 calc_en

Cette fonction est utilisable pour une ou deux molécules et calcule l'énergie potentielle d'un système donné en argument (*atoms*). Elle prend aussi un deuxième argument avec une valeur de 0 par défaut : *Print*. Si *Print* = 1, on affiche les valeurs des distances de liaisons et d'angles du système donné en argument. *Print* sera égal à 1 en fin de procédure de Minimisation, de cette façon, on pourra voir si les valeurs des angles et distances du système final sont proches ou égales aux valeurs de références (ce qui est normalement le cas pour un système correctement minimisé).

On calcule ici chaque terme de l'énergie potentielle soit $E_{liaisons}$ et E_{angles} pour un système à une molécule d'eau, et E_{vdv} et E_{elec} en plus pour un système à deux molécules d'eau.

2.4.3 derivate

Dérivation numérique de l'énergie potentielle en fonction de toutes les coordonnées selon un δ .

2.4.4 calc_GRMS

Calcule le GRMS avec la formule :

$$\sqrt{\frac{1}{n} \sum_{n=1}^n \left(\frac{\partial f}{\partial x_i} \right)^2} \quad (3)$$

On fait la somme de tous les éléments du gradient au carré, on divise par n qui représente le nombre d'élément dans le gradient (3 valeurs par atomes) et on fait la racine carré.

2.4.5 calc_decent

Procédure de Minimisation (steepest decent). On fixe certaines variables comme le seuil, λ , δ , le nombre de pas maximal...

Tant que le GRMS est **supérieur** au seuil et que le nombre de pas est différent du nombre de pas maximal, on effectue les étapes :

- Application de l'équation (1) du steepest decent et obtention de nouvelles coordonnées
- Calcul d'un gradient avec les nouvelles coordonnées
- Calcul d'un GRMS avec le nouveau gradient
- Calcul de l'énergie potentielle avec les nouvelles coordonnées

- Sauvegarde des valeurs de GRMS et d'énergie potentielle pour réaliser un graphique
- Affichage des valeurs d'énergie, GRMS et le nombre de pas effectué

Lorsque l'on sort de la boucle while, on procède à l'écriture de deux fichiers : un fichier pdb qui va contenir le système final avec une énergie minimale et un fichier png qui est un graphique représentant le GRMS et l'énergie potentielle en fonction du pas. Le nom de ces deux fichiers correspond au nom en argument 2 de la ligne de commande (argv[2]).

3 Résultats

Nous allons présenter ici une utilisation de notre programme avec une molécule d'eau puis avec deux molécules d'eau. On lance notre script avec la commande :

```
python SD.py water.pdb water_opti.pdb
```

Ce système provient du TP minimisation (le fichier de base a deux molécules d'eau, nous avons enlevé une des deux).

3.1 Une molécule d'eau

Voici la sortie en fin de minimisation pour une molécule d'eau :

```
Energie : 3.730075307928796e-07
GRMS : 0.008508971746795532
STEP : 20
```

```
Valeur pour la molécule 1 :
Liaison OH2-H1 : 0.9572223199463645
Liaison OH2-H2 : 0.9572159473097506
Angle : 1.8241931267207618
```

On constate qu'au bout de 20 pas, on obtient une énergie de l'ordre de 10^{-7} kcal/mol ce qui est très correcte, ainsi qu'un GRMS inférieur à 0.01, ce qui indique que nous sommes proche du minimum de la fonction d'énergie. Au niveau des valeurs de distances et d'angle du système finale, on voit que les valeurs sont très proches des valeurs de références utilisées comme valeurs à l'équilibre.

Pour la valeur à l'équilibre de la distance de la liaison O-H, nous avons utilisé la valeur de 0.9572 et pour la valeur de l'angle à l'équilibre la valeur de 1.824 radian.

Au niveau de la structure, voici la structure après la procédure de minimisation :

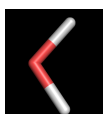


FIGURE 2 – Molécule d'eau avec une énergie potentielle minimale

La structure ne nous apportera pas beaucoup d'informations étant donné que nous avons qu'une seule molécule d'eau sans interaction possible avec d'autres atomes.

Le graphique obtenu pour le GRMS et l'énergie est le suivant :

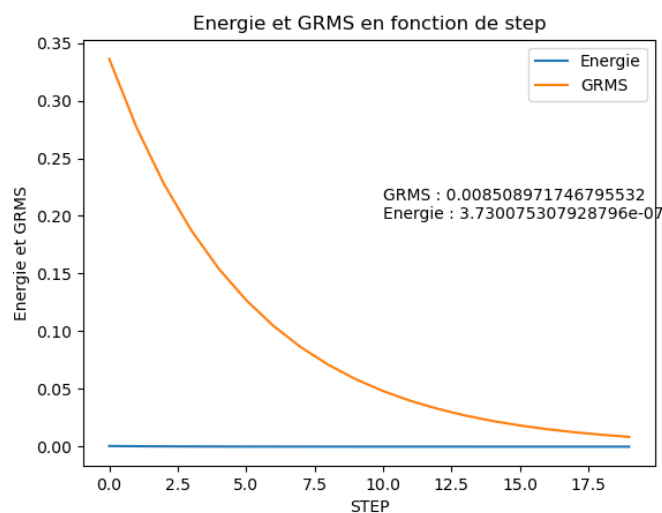


FIGURE 3 – Graphique de l'énergie et le GRMS en fonction du pas pour une molécule d'eau

On voit une bonne convergence du GRMS qui atteint très vite la valeur seuil.

3.2 Deux molécules d'eau

On lance la commande :

```
python SD.py water2.pdb water2_opti.pdb
```

Ce système provient du TP minimisation (c'est le même fichier que pour le système à une molécule d'eau mais cette fois, on a gardé les deux molécules dans le fichier pdb).

La sortie obtenue en fin de minimisation pour un système à deux molécules d'eau :

Energie : -6.694565525043924

GRMS : 0.1378817262626333

STEP : 10000

Valeur pour la molécule 1 :

Liaison OH2-H1 : 0.960150736902652

Liaison OH2-H2 : 0.9614702385929232

Angle : 1.8084091685688999

Valeur pour la molécule 2 :

Liaison OH2-H1 : 0.9549102044504822

Liaison OH₂-H₂ : 0.9758808755004569

Angle : 1.7897224576890953

LJ : 1.608440480709549

Coulomb : -8.55372113915169

Avec deux molécules d'eau, l'algorithme ne converge pas et atteint la limite de pas qui est de 10000. On obtient une énergie de -6.69 pour un GRMS de 0.13. La valeur d'énergie est assez bonne si on se réfère au TP Minimisation où l'algorithme de Steepest Decent avec un line search avait trouvé une énergie de -6.9256. On ne retrouve pas les valeurs de références pour les angles ni pour la valeurs de la longueur des liaisons. Cela s'explique par le fait que nous n'avons pas convergé.

Le graphique montre que les valeurs de GRMS et d'énergie chute très vite :

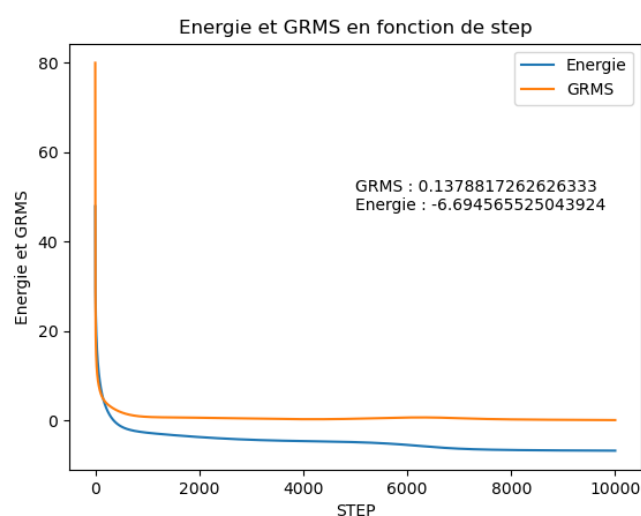


FIGURE 4 – Graphique de l'énergie et le GRMS en fonction du pas pour deux molécules d'eau

Au niveau des structures, à gauche la structure non minimisée, et à droite la structure minimisée :

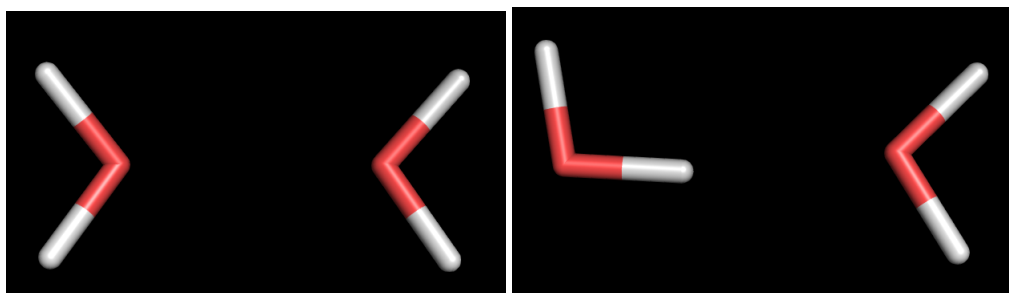


FIGURE 5 – Système de deux molécules d'eau non minimisée et minimisée

On constate que la molécule de gauche dans la structure minimisée (figure de droite) a effectué une rotation, par rapport à la molécule de gauche dans la structure initiale (figure de gauche), afin de minimiser l'énergie potentielle.

4 Conclusion

Nous avons réussi à obtenir un algorithme fonctionnant sur des systèmes d'une molécule d'eau et de deux molécules d'eau. Pour le système à une molécule d'eau, nous obtenons de bonnes valeurs d'énergies et de GRMS. Les valeurs finales sont proches des valeurs à l'équilibre. Pour le système à deux molécules d'eau, l'algorithme ne converge pas et on ne retombe pas exactement sur les valeurs de références pour les valeurs de liaisons et d'angles.

De plus, pour la minimisation de nos données nous n'utilisons pas d'algorithme de line search pour trouver un pas de minimisation optimal, il serait donc possible de réaliser une recherche de ce pas optimal dans le but d'améliorer la qualité de la minimisation.