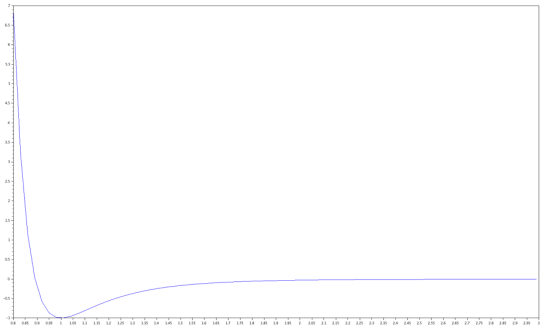
RO04 - Projet 2 - Optimisation non linéaire

Jingyi HU

October 31, 2018

1 Conformation atomique

1.1 Tracer le potentiel



Le potentiel de Lennard-Jones entre deux atomes distants de r est donné par: $V(r) = \frac{1}{r^{12}} - \frac{2}{r^6}$. Le premier terme décrit la tendance d'exclusion et le deuxième décrit l'interaction. Ils s'excluent mutuellement s'ils sont suffisamment proches. Au fur et à mesure qu'ils s'éloignent, l'attraction augmente d'abord et diminue ensuite. Lorsque les deux sont assez éloignés, le potentiel devient de plus en plus négligeable. Il atteint un minimum global(-1) dans l'équilibre entre deux tendance où r vaut 1.

Configurations théoriques minimales : La structure la plus stable de N points, c'est à dire la structure cristalline triangulaire avec tous les côtés de longueur 1, atteint un potentiel minimum. Ainsi le J optimal est égale au négatif la somme de longueurs de tous les côtés, c'est-à-dire le nombre de sommets de la structure.

N	Configuration théorique	J optimal
2	Un segment ($r = 1$)	-1
3	Un triangle équilatéral ($r = 1$)	-3
4	Cône triangulaire équilatéral ($r = 1$)	-6

1.2 Programmation sur Scilab

Dans un premier temps, on développe une fonction ***Lennardjones()*** pour calculer la valeur de J en donnant des coordonnées d'un point. À l'aide des fonctions ***CostLennardjones()*** et ***optim()***, on devient capable de trouver, à partir d'un point de départ et de manière de méthode de gradient, la solution optimale qui minimise J. Voici sont les résultats d'exécution pour $N = 2$ et 3 :

N	X de test	J	J opt
2	[1;0;0;0;0;0]	-1	-1
3	[1;0;0;0;1;0;0;0;1]	- 0.703125	-3

1.3 Solution optimale pour N = 4

Avec l'exemple de test, on a réussi à trouver que la valeur minimale de J vaut -6 où ces 4 composants construit un cône triangulaire équilatéral. On remarque bien que le résultat obtenu est identique par rapport à la valeur théorique (-6).

1.4 Solution optimale pour N = 8

Point changé	Temps d'exécution	$fopt$
X(donné)	5,39 s	-18.976056
Xopt	2,46 s	-18.976056
x2: (1,0,0) => (0.2,0,0)	5,19 s	-16.819321
x3: (1,1,0) => (0.2,0.2,0)	5,34 s	-16.01714

En premier lieux, on prend les points initiaux comme la position de départ, ensuite on la reinitialise par la configuration optimale (xopt) trouvée. Ces deux tentatives nous renvoient exactement les mêmes

solutions, surtout la deuxième tentative prend beaucoup moins du temps à retrouver la solution optimale. On pourrait dire que la valeur de J associée (**-18.976056**) est un minimum local.

Afin de vérifier si cela est aussi un optimum global, on pense à tirer aléatoirement la position initiale dans la deuxième étape, pour voir si différents points de départ peuvent toujours trouver une même solution optimale. On remarque une influence signifiante du choix des données initiales sur le résultat de l'optimiser. Nous avons des raisons de croire que cet optimisuer peut facilement tomber et être coincé dans les zones de minimaux locaux.

1.5 Lennard-Jones Pénalisé

En rajoutant un terme de pénalisation dans la fonction objective de minimisation, on reprogramme les fonctions et reteste les exemples de test avec différents N comme les exercices précédents. C'est évident que, à chaque fois, la valeur minimale de J reste identique, en revanche, X_1 est bien placé à l'origine des coordonnées (O). On peut dire que le coefficient de pénalité Λ est donc suffisamment important pour contraindre la position de X_1. Puisque la structure finale de même N, reste toujours stable, donc on limite à la fois la rotation et la position de toute la structure.

N	$fopt$	X_1
3	-3	(0.0000002, 6.784D-08, 0.0000002)
4	-6	(-0.0000005, -0.0000006, -0.0000007)
8	-18.976056	(-0.0000025, -0.0000011, -0.000003)

2 Programmation quadratique séquentielle(SQP)

2.1 Exécution et analyse

Cet algorithme est pour but de résoudre un problème d'optimisation (non convexe) sous contraintes, en le remplacant par une séquence de fonctions quadratiques (convexes). On initialise les contraintes sur X et la fonction objective, le solveur calculer automatiquement et nous retourner la solution, le nombre d'itération ainsi la valeur optimale. En l'exécutant, on manage à trouver la solution optimale **fopt** = **-14.843248** dans seulement **5** itérations. En outre, seulement les contraintes linéaires (1 à 4) sont actives ici, car larg(1:4) sont non nulls.

2.2 Réalisation de fonction

Nous avons programmé dans un premier temps une fonction de généralisation (***createAn***) de la matrice tri-diagonale $A_n = \frac{1}{h^2}$tridiag(-1,2,-1). Ensuite, on pense à identifier les variables d'entrée du solveur ***qpsolve(Q, p, C, b, ci, cs, me)*** comme le suivant :

me: Initialisé par 0, puisque l'on a que des contraites d'inégalité.

ci (**cs**): Limite inférieure de X est un vecteur colonne des valeurs infinies négatives (positives).

Autres: Q <= A_N', p <= -b_N, b <= d_N, C <= Matrice d'identité (diagonale) de dimension N.

2.3 Analyse des résultats

À travers ce solveur ***qpsolve()***, dans ***51*** itérations, une solution optimale ($xopt$) a été trouvé, ce qui minimise J par une valeur de ***-185.15346***. On présente ici les coordonnées de X en fonction de i comme le diagramme suivant :

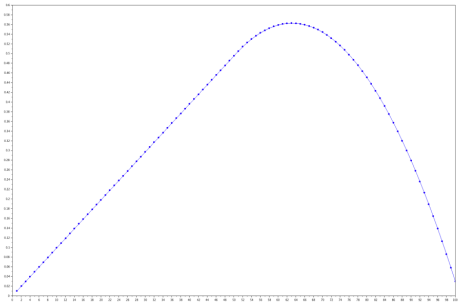


Figure 1: Coordonnées de X_i en fonctiond de i