

Komputerowe Metody Symulacji

Oskar Świtalski

9 października 2016

1 Wykład I

Prowadzący dr inż. Krzysztof Zborecki

Adres e-mail prowadzącego zborecki@if.pw.edu.pl

Strona z zadaniami <http://if.pw.edu.pl/~zborecki/KMS/download.html>

1.1 Program

- Metody deterministyczne, dynamika
- Metody stochastyczne
- Podstawy chemii kwantowej w użyciu do wł. mikro/nano układów (fizyki jądrowej/zderzenia ciężkich jonów)
- Omówienie algorytmów obliczeń z zbieżnych dziedzin

1.2 Literatura

- Podstawy symulacji komputerowych w fizyce – Hcerman
- Chemia kwantowa – Kołos
- Computential Physics – Koonin
- Computential Physics – Thijsen

1.3 Zaliczenie

- Laboratorium – 2 programy:
 - Mechanika molekularna
 - Rozwiązanie układu Schrödingera od czasu
- Kolokwium

1.4 Metody symulacji komputerowych

- model \rightarrow obserwable
- hamiltonian

- obserwable po zespole

$$\langle A \rangle = Z^{-1} \int_{\Omega} A(\vec{x}) f(\hat{\mathcal{H}}(\vec{x})) dx$$

gdzie

$$Z = \int_{\Omega} f(\hat{\mathcal{H}}(\vec{x})) dx - \text{f. rozdziału} / \text{suma statystyczna}$$

- obserwable po czasie

$$\vec{A}_t = (t - t_o)^{-1} \int_{t_o}^t A(x(\tau)) d\tau$$

- Hipoteza ergodyczna

1.5 Metody dynamiki molekularnej

Metody dynamiki molekularnej służą do obliczania trajektorii w przestrzeni fazowej zbioru molekuł, z których każda podlega klasycznym równaniom ruchu.

Komórka MD – komórka dynamiki molekularnej

$$\frac{du(t)}{dt} = K(u(t), t) \quad (1)$$

$$\hat{\mathcal{H}} = \sum_{i=1}^N \frac{p_i^2}{2m_i} + \sum_{K_j} U(u_{ij})$$

1.5.1 Efekty wymiaru (finite size effects)

Problem skończonej wielkości obszaru. Rozwiązanie warunki brzegowe.

$$A(\vec{x}) = A(\vec{x} + \vec{n} \cdot L)$$

$$\vec{n} = (n_1, n_2, n_3)$$

L – liniowe wymiary układu

Przy obliczaniu energii całkowitej ze wzoru (1) przyjmuje się tzw. konwencje najbliższego obrazu

$$r_{ij} = \min_n \{ |\vec{r}_i - \vec{r}_j + \vec{n}L| \} \quad (2)$$

cząstka oddziałuje tylko z $N-1$ cząstkami z komórki MD lub z jej najbliższego obszaru.

1.6 Schematy całkowania

MD jest zagadnieniem z warunkami początkowymi

$$m\dot{r} = p_i \quad (3)$$

$$\dot{p}_i = \sum_{i < j} F(x_j) \quad (4)$$

(4) wymaga $\frac{N(N-1)}{2}$ obliczeń jest to najbardziej złożona część obliczeniowa.

Algorytm taki który działa krok po kroku. U podstaw metod numerycznych służących do rozwiązywania równań różniczkowych stoi rozwinięcie w szereg Taylora.

$$u(t+h) = u(t) + \sum_i^{n-1} \frac{h^i}{i!} u^{(i)}(t) + R_n \quad (5)$$

h – krok dyskretyzacji

R_n – reszta

Dla $n = 2$

$$\frac{u}{t} = h^{-1} [u(t+h) - u(t)] + d(h) + o(h) \quad (6)$$

$$\frac{u}{t} = h^{-1} [u(t) - u(t-h)] + d(h) + o(h) \quad (7)$$

Algorytm Eulera

$$u(t+h) = u(t) + hK(u(t), t) \quad (8)$$

$$\dot{u} = K(u(t), t) \quad (9)$$

K - operator różniczkowy

Błąd - met. num.

Wybór kroku h w zależności od przyjętej wielkości fluktuacji energii potencjalnej.

1.7 obliczanie wielkości termodynamicznych

$$E = const.$$

$$T = const.$$

$$NVE$$

średnia po trajektorii w takim układzie

$$\vec{A} = \langle A \rangle_{NVE} = \lim_{n \rightarrow \infty} (t' - t_o)^{-1} \int_{t_o}^{t'} A(\vec{r}^N(t), \vec{p}^N(t), V) dt$$

$$\vec{E}_k = \lim_{\tau \rightarrow \infty} (t; -t_o)^{-1} \int_{t_o}^{t'} E_k(u(t)) dt$$

$$\vec{U} = \lim_{\tau \rightarrow \infty} (t; -t_o)^{-1} \int_{t_o}^{t'} U(V(t)) dt$$

Korzystając z zasady ekwipartycji energii:

$$\frac{mv_i^2}{2} = \frac{k_B T}{2}$$

$$E_c = E_K + U$$

1.8 Organizacja symulacji

- Inicjowanie – zadanie warunków początkowych – położenia i pędy
- Doprowadzenie do równowagi
- Realizacja według zadanego algorytmu

$$H = \frac{1}{2m} \sum_{i=1}^N p_i^2 + \sum_{i<j} u(r_{ij})$$

z równania Newtona

$$\frac{d^2 V_i(t)}{dt^2} = \frac{1}{m} \sum_{i<j} F_i(r_{ij})$$

1.9 Algorytm Verleta

Schemat rozwiązywania równania Newtona jest następujący:

$$\frac{d^2 r_i}{dt^2} = \frac{1}{h^2} [r_i(t+h) - 2r_i(t) + r_i(t-h)] = \frac{1}{m} F_i(t) \quad (10)$$

$$r_i(t+h) = 2r_i(t) - r_i(t-h) + \frac{F_i(t)h^2}{m} \quad (11)$$

$t-h, t \rightarrow t+h$ - metoda 2kierunkowa

Stosując oznaczenia

$$\begin{aligned} t_n &= nh \\ r_i^n &= r_i(t_n) \\ F_i^n &= F_i(t_n) \end{aligned}$$

$$r_i^{n+1} = 2r_i^n - r_i^{n-1} + \frac{F_i^n h^2}{m} \quad (12)$$

$$V_i^n = \frac{(r_i^{n+1} - r_i^{n-1})}{2h} \quad (13)$$

1.10 Algorytm Verleta NVE ma postać:

MD z alg Verleta

- określenie położenia r_i^0, r_i^1
- określenie sił dla n tego kroku F_i^n
- określenie położenia dla $n+1$ kroku czasowego zw względu (12)
- obliczenie prędkości dla $n+1$ -go kroku (13)

1.11 Przykład układu monoatomowego z oddziaływaniem typu oddziaływania L-J

$$U(r_{ij}) = 4\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right)$$

$$F_x = 48 \left(\frac{\epsilon}{\sigma^2} \right) (x_i - x_j) \left[\left(\frac{\sigma}{r_{ij}} \right)^{14} - \frac{1}{2} \left(\frac{\sigma}{r_{ij}} \right)^8 \right]$$

1.12

- układ z N at. jednakowych (argonu)
- przeprowadzić symulacje MD w ukł $E_z = \text{const.}$
- wylosować pędy z rozkładu maxwella
- wygenerować położenia w taki sposób aby atomy tworzyły komórkę elementarną kryształu gazu szlachetnego.
- histogramy pędów
- jmol - wizualizacja komórki
- E kin z rozkładu maxwella
- komórka elementarna(sześcian) jest w kuli at odbijają się od kuli
- przeskalowanie pędów tak by komórka/kryształ spoczywała $\vec{p}_i(0) = \vec{p}_i(0) = -\frac{1}{N} \sum_{i=1} N \vec{p}_i(0)$

potencjały i siły

- Oddziaływanie van Der Waleśa

$$V^P = \epsilon \left(\frac{R^{12}}{r_{ij}^{12}} - 2(R/r_{ij})^6 \right)$$

$$r_{ij} = |\vec{r}_i - \vec{r}_j|$$

$$V^s = 0, \text{ gdy } r_i < L$$

$$V^s = \frac{1}{2} f(r_i - L)^2, \text{ gdy } r_i \geq L$$

całkowity element potencjalny:

$$V = \sum_{i=1}^{N-1} \sum_{j=0}^{i-1} V^P(r_{ij}) + \sum_{i=1}^{N-1} V^s(r_{ij})$$

$$\vec{F}_i = \sum_{0 \leq j \leq N} \vec{F}_{i(j)}^P + \vec{F}_i^s$$

siły:

$$\vec{F}_{ij}^P = \frac{dV^P(r_{ij})}{d\vec{r}} = 12\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - 6 \left(\frac{\sigma}{r_{ij}} \right)^6 \right) \frac{\vec{r}_i - \vec{r}_j}{r_{ij}^2}$$

$$\vec{F}_i^s = -\frac{dV^s(x)}{d\vec{r}_i} = 0, r_i < L$$

$$\vec{F}_i^s = -\frac{dV^s(x)}{d\vec{r}_i} = f(L - r_i) \frac{\vec{r}_i}{r_i}, r_i \geq L$$

1.13 .xyz file

n atomów np 216 nazwa n razy: symbol atomu x_1 y_1 z_1 np Ar 1 1 1

1.14

- C/C++
- double
- nie używać funkcji pow - patrz fastpow
-

schemat numerycznych $H = \frac{1}{2m} \sum_{i=1}^N |p_i|^2 + V$ równania ruchu postać:

$$\dot{r}_i = dH/dp = p_i/m$$

$$\dot{p}_i = -dH/dr_i = \vec{F}_i$$

całkowanie z krokiem czasowym τ t - dany krok czasowy symulacji

r_i