

マルチスケール計算材料科学

原子系のシミュレーション (分子動力学法)

東京大学生産技術研究所 梅野宜崇

平成 24 年度夏学期

1 分子動力学法の基礎

原子 (または分子) の動きを追跡することによって物質の特性を評価するという試みはかなり以前よりなされてきた。そのためには膨大な量の数値計算をする必要があるため、計算機が用いられる。このような計算機による数値実験を計算機実験あるいはコンピュータシミュレーション (computer simulation) と呼んでいる。しかし、そのような手法の重要性が認識されだしたのはコンピュータの能力が著しく向上した近年のことである。今日、超大規模集積回路やマイクロマシンなどのように、原子オーダーの破損が信頼性に大きな影響を及ぼす非常に微小な機器の開発が進められていることなどから、工学材料の物性に関する原子レベルの研究が注目されている。ここでは、原子 (分子) シミュレーションの一つである分子動力学法 (Molecular Dynamics; MD) の基礎について解説する。以下に参考図書をあげておく。

- コンピュータシミュレーション, 上田顕, 朝倉書店
- コンピュータシミュレーションの基礎, 岡崎進, 化学同人
- Molecular Dynamics Simulation, Haile, John Wiley & Sons, Inc.
- The Art of Molecular Dynamics Simulation, Rapaport, Cambridge University Press

分子動力学法では、物質系をニュートン力学に従う原子からなる質点系と考える。それぞれの原子についての運動方程式を立て、それを数値積分することによって時々刻々の原子の位置と速度のデータを求めていく。時間 $t = t_0$ において、原子が図 1 左に示すような位置にあったとする。原子と原子の間には引力または斥力が働いているため、各原子には加速度がかかるが、これは原子の位置などから計算できる。また、微小時間 Δt 前の原子の位置もわかっているとす。この 3 つの情報により、運動方程式を数値積分することがで

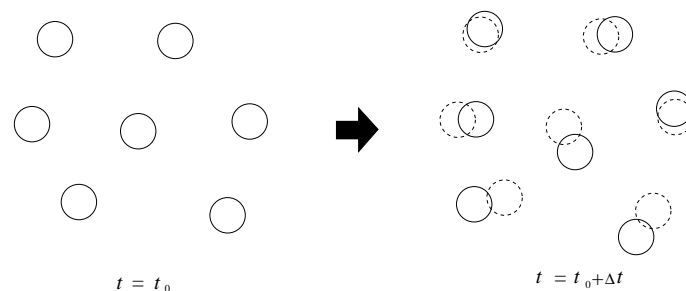


Fig. 1 原子を時間ステップ Δt ずつ動かす

き, 微小時間 Δt 後の原子の位置が計算できる (図 1 右). (この Δt を時間ステップといい, 時間の刻み幅を表す.) この操作が, MD 法でいう「原子を動かす」ことである. $t = t_0 + \Delta t$ での原子の位置がわかれば, その時刻での原子に働く力も計算でき, さらに Δt 後の原子の位置が求められる. この作業を繰り返すことによって原子の運動を追跡してゆくのが MD 法である.

2 運動方程式と数値積分

では, 実際に運動方程式を数値積分する方法を簡単な例で示してみよう.

2.1 Verlet 法

他の原子から力を受けながら運動する N 個の原子からなる質点系を取り扱うとする. 原子は同種でなくても良いのだが, 簡単のために質量 m の同種原子とする. 原子 i の受ける力を F_i とすれば, 運動方程式は

$$\frac{d^2 \mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m}, \quad i = 1, 2, \dots, N \quad (1)$$

である. 3次元空間の運動では, 上式は $3N$ 個の連立2階常微分方程式となる. この式は一般に解析的には解けないので, 数値積分する必要がある.

微分方程式を数値積分するには, 微分方程式を差分方程式で近似する. これは, 大雑把に言えば時間微分を

$$\frac{d\mathbf{r}}{dt} \simeq \frac{1}{\Delta t} \{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)\} \quad (2)$$

と近似することに相当する. Verlet 法では, 原子 i の位置 \mathbf{r}_i と速度 \mathbf{v}_i とを

$$\mathbf{r}_i(t + \Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + (\Delta t)^2 \frac{\mathbf{F}_i(t)}{m} \quad (3)$$

$$\mathbf{v}_i(t) = \frac{1}{2\Delta t} \{\mathbf{r}_i(t + \Delta t) - \mathbf{r}_i(t - \Delta t)\} \quad (4)$$

によって求める. この式から, とびとびの離散時刻 $n\Delta t$ ($n = 1, 2, 3, \dots$) における原子の位置と速度の値が求められる. なお, Δt は普通, 1fs ~ 数 fs (1fs = 10^{-15} s)にとられる.

2.2 速度 Verlet 法

実際のプログラミングでは, 数値計算上の桁落ちを防ぐため, 数学的には等価であるが以下のように変形された式を用いる. 式 (3) に $(1/2)\mathbf{r}_i(t - 2\Delta t)$ を足して引いて,

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \frac{1}{2}\mathbf{r}_i(t) + \frac{1}{2}\mathbf{r}_i(t) \\ &\quad - \mathbf{r}_i(t - \Delta t) + \frac{1}{2}\mathbf{r}_i(t - 2\Delta t) - \frac{1}{2}\mathbf{r}_i(t - 2\Delta t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t) \\ &= \mathbf{r}_i(t) + \Delta t \left\{ \mathbf{v}_i(t - \Delta t) + \frac{\Delta t}{m_i} \frac{\mathbf{F}_i(t) + \mathbf{F}_i(t - \Delta t)}{2} \right\} + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t) \end{aligned} \quad (5)$$

となる. 時刻 t における速度を

$$\mathbf{v}_i(t) = \mathbf{v}_i(t - \Delta t) + \frac{\Delta t}{m_i} \frac{\mathbf{F}_i(t) + \mathbf{F}_i(t - \Delta t)}{2} \quad (6)$$

で求めることにすれば, 時刻 $t + \Delta t$ における原子の位置は

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t) + O(\Delta t^4) \quad (7)$$

により求められる.

2.3 Gear の予測子・修正子法

$$\mathbf{r}_n(t) = \frac{\Delta t^n}{n!} \frac{d^n \mathbf{r}(t)}{dt^n} \quad (8)$$

とおくと、テイラー展開より例えば 5 次までとると

$$\begin{aligned} \mathbf{r}_0^p(t + \Delta t) &= \mathbf{r}_0(t) + \mathbf{r}_1(t) + \mathbf{r}_2(t) + \mathbf{r}_3(t) + \mathbf{r}_4(t) + \mathbf{r}_5(t) + O(\Delta t^6) \\ \mathbf{r}_1^p(t + \Delta t) &= \mathbf{r}_1(t) + 2\mathbf{r}_2(t) + 3\mathbf{r}_3(t) + 4\mathbf{r}_4(t) + 5\mathbf{r}_5(t) + O(\Delta t^6) \\ \mathbf{r}_2^p(t + \Delta t) &= \mathbf{r}_2(t) + 3\mathbf{r}_3(t) + 6\mathbf{r}_4(t) + 10\mathbf{r}_5(t) + O(\Delta t^6) \\ \mathbf{r}_3^p(t + \Delta t) &= \mathbf{r}_3(t) + 4\mathbf{r}_4(t) + 10\mathbf{r}_5(t) + O(\Delta t^6) \\ \mathbf{r}_4^p(t + \Delta t) &= \mathbf{r}_4(t) + 5\mathbf{r}_5(t) + O(\Delta t^6) \\ \mathbf{r}_5^p(t + \Delta t) &= \mathbf{r}_5(t) + O(\Delta t^6) \end{aligned} \quad (9)$$

と書ける。添字 p はテイラー展開から求めたままの値，すなわち予測値 (predictor) であることを示す。ここで

$$\Delta \mathbf{r}_2 = \frac{\Delta t^2}{2!m} \mathbf{F}\{\mathbf{r}_0^p(t + \Delta t)\} - \mathbf{r}_2^p(t + \Delta t) \quad (10)$$

と定義する。もし予測値が正しければ $\Delta \mathbf{r}_2 = 0$ となるがこれは一般には成り立たないので，この分を次のように修正してやる。

$$\mathbf{r}_n^c(t + \Delta t) = \mathbf{r}_n^p(t + \Delta t) + f_n^{(2)} \Delta \mathbf{r}_2 \quad (11)$$

添字 c は修正子 (corrector) を表す。 $f_n^{(2)}$ の係数についてはテイラー展開の次数や微分方程式の次数 m などに依存し，最適なものがギアによって示されている。一例を表に示す。

Table 1 Coefficients for predictor-corrector method.

	0	1	2	3	4	5
$\dot{\mathbf{r}} = f(\mathbf{r})$	$\frac{251}{720}$	1	$\frac{11}{12}$	$\frac{1}{3}$	$\frac{1}{24}$	
$\ddot{\mathbf{r}} = f(\mathbf{r})$	$\frac{3}{20}$	$\frac{251}{360}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$
$\ddot{\mathbf{r}} = f(\mathbf{r}, \dot{\mathbf{r}})$	$\frac{3}{16}$	$\frac{251}{360}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$

3 境界条件

MD は，原子一つ一つの運動を追跡する方法であり，取り扱う原子の数に従って計算量が増える。最高では数千万個の原子系のシミュレーションなども行われているが，通常は取り扱える原子数は数千から数万個である。このような小数の原子系でマクロな系の性質を調べることが可能なのだろうか。これを解決するため，以下に説明する周期境界条件 (periodic boundary condition) が用いられる。これはかなり強引な方法のように見えるが，周期境界条件のもとで運動する数百個程度の原子系の示す熱力学的性質がマクロな系のそれをよく表すことが確認されていることなどから，MD ではごく自然に導入される。

N 個の原子が一辺 L の立方体の容器内で運動しているとする。周期境界条件では，原子は立方体の壁から力を受けることなくそのまま容器の外に出てしまい，対面の相対する位置から同じ速度で入ってくるとする。この状況を 2 次元平面内の運動について示すと図 2 のようになる。中央の立方体 (図では正方形) を注

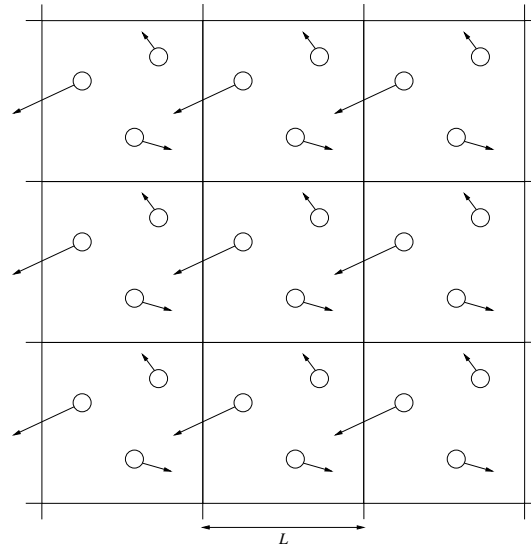


Fig. 2 周期境界条件での基本セル (中央) とイメージセル中の原子の運動

目する体系とし, この立方体を基本セル (primary cell) と呼ぶ. また他の前後左右上下に積み重ねた, 基本セル以外のセルをイメージセル (image cell) またはレプリカセル (replica cell) と呼ぶ. 図に示したように, 基本セル内の原子の運動と全く同じ運動が全てのイメージセルで繰り返されているので, 基本セルから飛び出した原子は相対する壁から入ってくることになる.

4 MDにおけるアンサンブル

以上の方法では, 取り扱う系の原子数 N , 系の体積 V , 系のエネルギー E が一定である小正準集合 (ミクロカノニカルアンサンブル) のシミュレーションを行うことになる. この系を NVE アンサンブルという. しかし, 実際には温度や圧力が一定の条件下における物性を知りたいことが多い. そこで, 圧力や温度をパラメータとして入力すればその値へ向かって系が自動的に緩和するように, 体系を力学系として拡張する方法が開発されている. ここでは, 温度一定 (NTV アンサンブル) と, 圧力一定 (NPH アンサンブル) の 2 つの MD 法について述べる. この 2 つの方法をまとめて圧力と温度を同時に制御することも容易にできる.

4.1 速度スケールリング法による温度制御

温度を一定に保つ簡単な方法は, 速度スケールリング法と呼ばれる, 原子の速度を強制的に操作する方法である. 原子 i の運動量を p_i とすると, 系の温度 T は

$$T = \frac{1}{3Nk_B} \sum_i \frac{p_i^2}{m} \quad (k_B \text{はボルツマン定数}) \quad (12)$$

で与えられるので, 目的の温度を T_0 とすれば, 各原子の速度を $\sqrt{T_0/T}$ 倍するという操作を何度か繰り返せばよい. 目的の温度の制度が数%でよければ, この方法は十分実用的である.

4.2 Nosé-Hoover 法による温度制御

もっと理論的にはっきりした温度制御の方法としては, Nose-Hoover 法がある. これは, 原子系を温度 T_0 の熱浴と熱平衡にある体系と考え, 熱浴とのエネルギーの授受を考慮する方法である. 原子系にこのエネル

ギ-授受を担う付加自由度 s を導入し, 温度 T のカノニカル分布を実現する力学系の定式化が能勢らによってなされているが, 説明が難しくなるのでここでは省略する. 温度 T は系の内部エネルギーが全運動エネルギーに等しいとして

$$T = \frac{2}{3Nk_B} \langle E \rangle \quad (13)$$

$$E = \sum_i \frac{1}{2} m v_i^2 \quad (14)$$

で与えられる. Nose-Hoover の方法 [J.Chem.Phys.81,p.511 (1984)] では, 原子系に温度制御のための熱浴を設けた拡張した力学系を考える. この熱浴と原子の間でエネルギーをやりとりさせることによって T を一定に保つ.

原子系にエネルギー授受を担う付加的な自由度 s を導入する. 対象とする原子の位置と運動量を現実の量で示したものを現実系と呼び, 新しい自由度 s を用いて

$$dt = \frac{dt'}{s} \quad (15)$$

によって規格した量で示したものを仮想系と呼ぶことにする. 後者の位置ベクトルと運動量ベクトルに ' をつけて区別すると現実系と仮想系の変数の間には

$$\mathbf{r}_i = \mathbf{r}'_i \quad (16)$$

$$\mathbf{p}_i = \frac{\mathbf{p}'_i}{s} \quad (17)$$

なる関係が成立する. 温度制御のための熱浴を含めた拡張系におけるハミルトニアン H_2 は

$$H_2 = \sum_i \frac{\mathbf{p}'_i{}^2}{2ms^2} + \sum_i \sum_{j>i} \phi(r'_{ij}) + \frac{P_s^2}{2Q} + 3Nk_B T \log s \quad (18)$$

となる. また, P_s は s に正準共役な運動量であり, Q は s の質量に相当する任意定数である. これによって, 設定した温度 T のカノニカル分布を実現することができる. 仮想系の正準方程式は,

$$\frac{d\mathbf{r}'_i}{dt'} = \frac{\partial H_2}{\partial \mathbf{p}'_i} = \frac{\mathbf{p}'_i}{ms^2} \quad (19)$$

$$\frac{d\mathbf{p}'_i}{dt'} = -\frac{\partial H_2}{\partial \mathbf{r}'_i} = -\sum_{i \neq j} \frac{\phi'}{r'_{ij}} (\mathbf{r}'_i - \mathbf{r}'_j) \quad (20)$$

$$\frac{ds}{dt'} = \frac{\partial H_2}{\partial P_s} = \frac{P_s}{Q} \quad (21)$$

$$\frac{dP_s}{dt'} = -\frac{\partial H_2}{\partial s} = \frac{1}{s} \left[\sum_i \frac{\mathbf{p}'_i{}^2}{ms^2} - 3Nk_B T \right] \quad (22)$$

のように導かれる. これを, 現実系の変数を用いて書き直すと,

$$\frac{d\mathbf{r}_i}{dt} = \frac{\mathbf{p}_i}{m} \quad (23)$$

$$\frac{d\mathbf{p}_i}{dt} = -\sum_{i \neq j} \frac{\phi'}{r_{ij}} (\mathbf{r}_i - \mathbf{r}_j) - \zeta \mathbf{p}_i \quad (24)$$

$$\frac{ds}{dt} = s\zeta \quad (25)$$

$$\frac{d\zeta}{dt} = \frac{1}{Q} \left[\sum_i \frac{\mathbf{p}_i^2}{m} - 3Nk_B T \right] \quad (26)$$

となる．ここで，

$$\zeta = \frac{P_s}{Q} \quad (27)$$

である．これが，熱浴の影響を付加した基本セル内原子の運動方程式であり，本式を解くことによって温度一定下の原子運動を直接得ることができる．

4.3 圧力制御法

図3左のように，シリンダーとピストンからなる容器に気体がつまっているとしよう．系の圧力は，ピストンが右に動くとき下がり，左に動くとき上がる．こうして系の体積を変えて望む圧力の状態におくことができる．圧力一定のMD法では，このシリンダーとピストンの系にならって，右図のように体積の変化するセルに原子が入っているような系を考える．すなわち， N 個の原子が一边 L の立方体にはいっているとし，系の体積 $V(= L^3)$ を新しい自由度としてもつ拡張された力学系を扱うのである．このとき，ピストンとは異なり，系は立方体の形状を保ったまま等方的に膨張・収縮する．この方法はセルの形を保って大きさだけを変化させるもので，Andersenの方法と呼ばれるものである．この他にセルの形状の変化を許す圧力制御法(Parrinello-Rahman法)も考案されている．

圧力 P はビリアル定理より，

$$P = \frac{Nk_B T}{V} - \frac{1}{3V} \left\langle \sum_i \sum_{j>i} r_{ij} \frac{d\phi(r_{ij})}{dr_{ij}} \right\rangle \quad (28)$$

により原子系から評価できる．ここで， N は系の原子数， k_B はボルツマン定数， V は系の体積， m は原子の質量， r_{ij} は原子 i と原子 j の距離， ϕ は原子間の対ポテンシャルであり， $\langle \rangle$ は長時間平均を表す．Parrinello-Rahmanの方法[J.Appl.Phys.52,p.7182(1981)]では，シミュレーションする系の形状を変化させて，圧力を制御する．基本セルにリザーバを付加した拡張した力学系を考える．基本セルが，基本ベクトル a, b, c を3辺とする直方体であるとし，このベクトルからなる行列を

$$h = \{a, b, c\} \quad (29)$$

とする．ここで，

$$r_i = h s_i \quad (30)$$

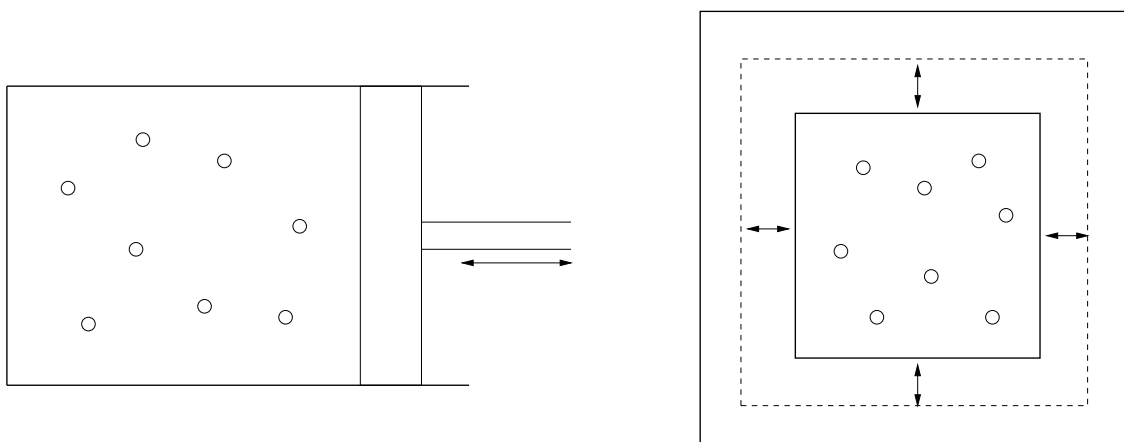


Fig. 3 圧力制御のモデル

より求められる原子 i の位置ベクトル r_i を基本ベクトル h で規格化したベクトル s_i を導入する．この時，原子 i の速度 v_i は，

$$v_i = \frac{dr_i}{dt} = \frac{d(hs_i)}{dt} = h\dot{s}_i + \dot{h}s_i \quad (31)$$

となる．ここで， $\dot{}$ は時間微分を表す． h が時間とともに変動することにより，系の体積や形状を変化させて圧力を一定にする． h を付加的な自由度として持つ拡張された力学系におけるハミルトニアン H_1 は，

$$H_1 = \frac{1}{2} \sum_i m v_i^2 + \sum_i \sum_{j>i} \phi(r_{ij}) + \frac{1}{2} W T_r \dot{h}^t \dot{h} + P \|h\| \quad (32)$$

となる．ここで， W は圧力制御のためのリザーバの質量に相当する任意定数である（32）式の第1項と第2項は基本セル内原子の運動エネルギーとポテンシャルエネルギーであり，第3項と第4項はリザーバの運動エネルギーと基本セルに対する仕事である．また， P は制御する系の圧力設定値であるが，この値が負の場合は3軸引張りを表すというように拡張して考えることができる．すなわち，系に働く静水圧応力を σ_h とすれば，

$$\sigma_h = -P \quad (33)$$

である．この H_1 より，正準形の運動方程式を書き下す．まず， s_i, h に正準共役な運動量をそれぞれ π_i, γ とすれば，

$$\pi_i = \frac{\partial H_1}{\partial \dot{s}_i} = m h^t \dot{h} s_i \quad (34)$$

$$\gamma = \frac{\partial H_1}{\partial \dot{h}} = W \dot{h} \quad (35)$$

である．これより，ハミルトンの正準方程式

$$\frac{ds_i}{dt} = \frac{\partial H_1}{\partial \pi_i} = \frac{1}{m} G^{-1} \pi_i \quad (36)$$

$$\frac{d\pi_i}{dt} = -\frac{\partial H_1}{\partial s_i} = -\sum_{i \neq j} \frac{\phi'}{|hs_{ij}|} G(s_i - s_j) \quad (37)$$

$$\frac{dh}{dt} = \frac{\partial H_1}{\partial \gamma} = \frac{\gamma}{W} \quad (38)$$

$$\frac{d\gamma}{dt} = -\frac{\partial H_1}{\partial h} = \sum_i \frac{1}{m} (h^{t-1} \pi_i \times h^{t-1} \pi_i) h^{t-1} - \frac{\phi'}{|hs_{ij}|} (s_{ij} \times s_{ij}) - P \|h\| h^{t-1} \quad (39)$$

が得られる．ここで，

$$G = h^t h \quad (40)$$

であり， \times はダイアド

$$\mathbf{x} \times \mathbf{x} = \begin{pmatrix} x_1 x_1 & x_1 x_2 & x_1 x_3 \\ x_1 x_2 & x_2 x_2 & x_2 x_3 \\ x_1 x_3 & x_2 x_3 & x_3 x_3 \end{pmatrix} \quad (41)$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (42)$$

を表す．式(36)(37)は，実座標に書き直すと

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{h} \frac{d\mathbf{s}_i}{dt} + \frac{d\mathbf{h}}{dt} \mathbf{s}_i = \frac{1}{m} \mathbf{p}_i + \frac{1}{W} \gamma \mathbf{h}^{-1} \mathbf{r}_i \quad (43)$$

$$\frac{d\mathbf{p}_i}{dt} = \mathbf{h}^{t-1} \frac{d\boldsymbol{\pi}_i}{dt} + \frac{d\mathbf{h}^{t-1}}{dt} \boldsymbol{\pi}_i = - \sum_{i \neq j} \frac{\phi'}{r_{ij}} (\mathbf{r}_i - \mathbf{r}_j) - \frac{1}{W} \mathbf{h}^{t-1} \mathbf{h}_v^t \mathbf{p}_i \quad (44)$$

が得られる．これが，リザーバの影響を付加した基本セル内原子の運動を律則する支配方程式(運動方程式)である．セル内原子の原子間力だけに基づく Newton の運動方程式に変えて，式(38)および式(39)と連動した本式を解くことにより，系の圧力制御下における原子運動を直接シミュレーションすることができる．

5 圧力・温度制御法

圧力一定条件下の運動方程式(式(38)(39)(43)(44))と温度一定条件下の運動方程式(式(23)(24)(25)(26))を組み合わせると，次のような圧力・温度一定条件の運動方程式が得られる．

$$\frac{d\mathbf{r}_i}{dt} = \frac{1}{m} \mathbf{p}_i + \frac{1}{W} \gamma \mathbf{h}^{-1} \mathbf{r}_i \quad (45)$$

$$\frac{d\mathbf{p}_i}{dt} = - \sum_{i \neq j} \frac{\phi'}{r_{ij}} (\mathbf{r}_i - \mathbf{r}_j) - \zeta \mathbf{p}_i - \frac{1}{W} \mathbf{h}^{t-1} \mathbf{h}_v^t \mathbf{p}_i \quad (46)$$

$$\frac{ds}{dt} = s\zeta \quad (47)$$

$$\frac{d\mathbf{h}}{dt} = \frac{\gamma}{W} \quad (48)$$

$$\begin{aligned} \frac{d\gamma}{dt} = & \sum_i \frac{1}{m} (\mathbf{p}_i \times \mathbf{p}_i) \mathbf{h}^{t-1} - \sum_i \sum_{j>i} \frac{\phi'}{r_{ij}} (\mathbf{r}_{ij} \times \mathbf{r}_{ij}) \mathbf{h}^{t-1} \\ & - P \|\mathbf{h}\| \mathbf{h}^{t-1} \end{aligned} \quad (49)$$

$$\frac{d\zeta}{dt} = \frac{1}{Q} \left[\sum_i \frac{\mathbf{p}_i^2}{m} - 3Nk_B T \right] \quad (50)$$

6 ポテンシャル関数

MDシミュレーションの結果を左右する大きな要因の一つは，原子間ポテンシャルである．現実の原子が感じているポテンシャルを厳密に求める事は非常に難しく，シミュレーションには何らかの近似を施して計算しやすくしたポテンシャル関数を用いる．ポテンシャルをより正確に記述しようとする，それだけ式が複雑になり，計算時間などの点で限界が生じる．したがって，対象とする系や明らかにしたい物性に適したポテンシャル関数を採用する事も重要である．シミュレーションを行う際には，これらのことを考慮してポテンシャル関数を選択する．

MDに用いられる原子間ポテンシャルは様々なものがあり，全てを網羅することはできないが，ここでは固体材料の計算によく使われるもののうちいくつかを例としてあげる．

なお，原子間の相互作用は距離が大きくなると限りなくゼロに近づくと考えられるので，いずれのポテンシャルにおいても適当なカットオフ距離 r_c を設け， $r > r_c$ については計算しないとするのが通常である(そうしないと計算量が無限大になる)．

6.1 2体ポテンシャル

まずもっとも簡単なのは、原子間ポテンシャルを原子間距離あるいは結合角などの関数で表示したものである。原子が2個ずつ単独で存在するときのポテンシャルエネルギー $\phi(r)$ の単純な総和で系全体のポテンシャルエネルギーが記述できるとき、 $\phi(r)$ を対ポテンシャルまたは2体間ポテンシャルという。すなわち、 N 個の原子系の場合、系全体のポテンシャルエネルギー E_{pot} は

$$E_{\text{pot}} = \frac{1}{2} \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N \phi(r_{ij}) \quad (51)$$

である (r_{ij} は原子 i と原子 j の距離)。このとき、ある原子に働く力は、それをとりまく原子がそれぞれ単独に存在するときの働く力をたしあわせたものとなる。また、ある程度以上原子間の距離が離れると両者の間に働く力は無視できる程小さくなるので、力及びポテンシャルの計算は r_c 以上離れた原子の影響は無視し (この r_c を切断距離 (cut-off radius) という)、切断距離内の原子についてのみ行う。

ϕ の関数形としては、金属などによく用いられるモース (Morse) ポテンシャル

$$\phi(r) = \varepsilon \{ \exp[-2\alpha(r - r_0)] - 2 \exp[-\alpha(r - r_0)] \} \quad (52)$$

(α, ε, r_0 は定数) や、ファンデルワールス力をよく再現するレナード-ジョーンズポテンシャル

$$\phi(r) = \varepsilon \left\{ \left(\frac{\sigma}{r} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^6 \right\} \quad (53)$$

(ε, σ は定数) 等がある。

2体ポテンシャルの利点の一つに、関数形が簡単であり多元系にフィッティングしやすいことがあげられる。例えば A 原子と B 原子を含む2元系では、A-A, B-B, A-B それぞれのペアによってパラメータを変えることでそれぞれの相互作用を変化させる。

7 3体ポテンシャル

2体ポテンシャルは2原子間のポテンシャルが2原子の距離だけの関数として決まると近似したものであるが、3体ポテンシャルでは原子間距離のみでなく、3原子の張る角度 (結合角) をも含んだ関数として表現される。代表的なものに、S-W (Stillinger-Weber) ポテンシャル (溶融のエントロピーや液体の構造因子などをよく再現する)、Tersoff ポテンシャル (主としてシリコン系のシミュレーションに非常によく用いられる) およびその改良型 (炭化水素系向けに開発された Brenner ポテンシャルはカーボンナノチューブのシミュレーションにもよく用いられる)、B-H (Biswas-Hanmann) ポテンシャル (アモルファス構造などをよく再現する) などがある。(なお、Tersoff ポテンシャルについては「発見的な Bond order ポテンシャル (後述)」という表現をされることがある。)

例として、Tersoff ポテンシャルの関数形を示す。

$$\Phi = \frac{1}{2} \sum_i \sum_{j \neq i} u_{ij} \quad (54)$$

$$u_{ij} = f_{ij} A_{ij} \exp(-\lambda_{ij} r_{ij}) - f_{ij} b_{ij} B_{ij} \exp(-\mu_{ij} r_{ij}) \quad (55)$$

$$b_{ij} = \left[1 + \left(\beta_i \sum_{k \neq i, j} \zeta_{ijk} \right)^{n_i} \right]^{-\frac{1}{2n_i}} \quad (56)$$

$$\zeta_{ijk} = f_{ik} \exp[\mu_{ij}^{m_i} (r_{ij} - r_{ik})^{m_i}] \left\{ 1 + \frac{c_i^2}{d_i^2} - \frac{c_i^2}{d_i^2 + (h_i - \cos \theta_{kij})^2} \right\} \quad (57)$$

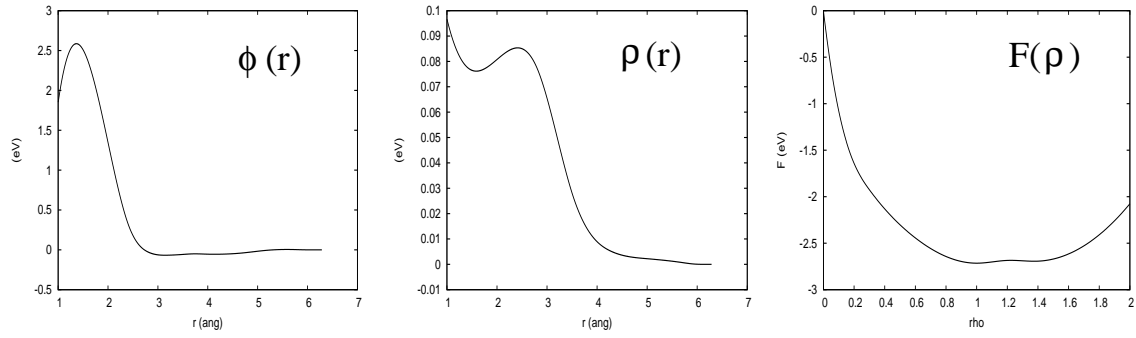


Fig. 4 EAM potential functions by Mishin et al.

θ_{kij} は角 kij , すなわち ik と ij のなす角を表す. $f_{ij} = f(r_{ij})$ はカットオフ関数であり, $r \leq R$ のとき 1, $r \geq S$ のとき 0, $R < r < S$ のとき

$$f = \frac{1}{2} + \frac{1}{2} \cos \left[\frac{\pi(r - R)}{S - R} \right] \quad (58)$$

である. $A, B, \lambda, \mu, m, c, d, h, S, R$ はパラメータ (定数) であり, 材料や求めたい物性値によって様々なものが提案されている.

7.1 多体ポテンシャル

上記の関数表現されたポテンシャルではバルク中の原子間力をうまく表現できるが, 電子密度分布がバルクとは著しく異なる表面や粒界, あるいは異相界面, 結晶欠陥部での原子に働く力を求めるのには不十分である. そこで Baskes らは, 金属では電子雲中に原子核 (イオン) が埋め込まれていると考え, その原子核 (イオン) を挿入する際のエネルギーを考慮に入れたポテンシャルを考案した. すなわち, 系の全エネルギー E_{tot} は次式のように電子密度依存項と非依存項の和で表される.

$$E_{tot} = \frac{1}{2} \sum_{i \neq j} \phi_{ij}(r_{ij}) + \sum_i F_i^{EAM}(\bar{\rho}_i) \quad (59)$$

ここで, $F_i^{EAM}(\bar{\rho}_i)$ は電子密度 $\bar{\rho}_i$ を持つ位置にある原子の埋め込みエネルギーを, $\phi_{ij}(r_{ij})$ は原子間の中心斥力を表す. また, 電子密度は

$$\bar{\rho}_i = \sum_{j \neq i} \rho(r_{ij}) \quad (60)$$

与えられる. ϕ, ρ, F は材料によって様々なものが提案されている. 図 4 に, Mishin らによる Al に対する関数を示す.

8 実際の計算における technical matters

8.1 カットオフ距離

どのような原子間ポテンシャル関数を使うにしても, 一般に, 原子間距離が大きくなるにしたがい相互作用は弱くなる. 一定距離以上離れた原子間の相互作用は無視できるとし, 計算を省略する必要がある (そうでないと, 周期境界条件を課した場合には無限回の計算を行わねばならなくなる). そこで, カットオフ距離 r_c を設定し, $r_{ij} > r_c$ なる原子対については相互作用の計算をしない, という方法が通常とられる. 平方根の計算を省略するため, r_{ij}^2 と r_c^2 の大小関係で判定するのがよい.

8.2 Bookkeeping 法 (neighbor list の利用)

原子 i に働く力やエネルギーを計算する際、そのまわりの原子をすべて探索し、カットオフ半径の球内に入っているかを判別する作業を行うが、これを毎ステップ行うと無駄が多い。なぜなら、 r_c より充分離れた距離にある原子は、しばらくの間はカットオフ球内に入ってくることはないため、いちいち距離（またはその 2 乗）を計算する必要がないからである。

そこで、 r_c よりも少し大きめのカットオフ距離 R_c を設定し、ある時間 t において、原子 i から半径 R_c の球内にある原子をリストアップしておく（neighbor list テーブルを作って記憶しておく）。時間 t から $t + \delta t$ までは原子 i に働く力・エネルギーはこのリストに登録された原子のうち、距離 r_c 以内のもののみ計算する。このような方法を Bookkeeping method（粒子登録法）と呼ぶ。

リストテーブルの更新の計算量は $O(N^2)$ であるのに対し、リストテーブルを使っの毎回のポテンシャル計算は $O(N)$ で済むことから、 R_c とリスト更新までのステップ数 n を適切に選べば、計算結果に影響を及ぼすこと無く計算の高速化が可能となる。目安としては、原子の速さの平均を \bar{v} として

$$R_c > r_c + \bar{v}n\Delta t \quad (61)$$

と取るとよい。Bookkeeping 法のプログラム例（3 方向周期境界を想定）を以下に示すが、やり方はいろいろあるので工夫のしどころである。

```
do i=1,n
  alistnum(i) = 0
  do j=1,n
    xx = rx(j)-rx(i); yy = ry(j)-ry(i); zz = rz(j)-rz(i)
    ix = 0; iy = 0; iz = 0;
    if (xx < -rrc) xx = xx + lx
    if (xx > rrc) xx = xx - lx
    if (yy < -rrc) yy = yy + ly
    if (yy > rrc) yy = yy - ly
    if (zz < -rrc) zz = zz + lz
    if (zz > rrc) zz = zz - lz
    dist2 = xx*xx + yy*yy + zz*zz
    if (dist2 < rc2) then
      alistnum(i) = alistnum(i)+1
      alist(i,alistnum(i),1) = j
      alist(i,alistnum(i),2) = ix
      alist(i,alistnum(i),3) = iy
      alist(i,alistnum(i),4) = iz
    endif
  enddo
enddo
```

8.3 領域分割法

Bookkeeping 法で、そのままではリストテーブルを更新する際に $n(n-1)$ の原子対について原子間距離の自乗の計算が発生する。系が大きくなるとこれが律速となってしまふ。そこで、シミュレーションセルを

多数の直方体ブロックに分割し，そのブロックをナンバリングしておく．ブロックの大きさを適切に設定しておけば，原子対の探索は隣接するブロックまで行えばよいようにできる．これを利用すれば，自乗の計算回数を大幅に減らすことが可能となり，高速化につながる．

9 演習 (1): 原子 3 個の MD 計算

- (1) 原子 (Al とする) を 3 個，適当な間隔で xy 平面上に配置せよ．このとき，Al 単結晶 (fcc 構造) の格子定数 a_0 は 4.05\AA ，再近接原子間距離は $a_0/\sqrt{2}$ であることを参考にせよ．
- (2) 配置した原子に働く力を計算せよ．原子間ポテンシャルには式 (52) の Morse ポテンシャルを用い，パラメータ ε, α, r_0 はそれぞれ 0.27 eV , 1.16 \AA^{-1} , 3.25 \AA とせよ．
- (3) 初期速度をゼロとし，時間ステップ $\Delta t = 1\text{ fs}$ ($=10^{-15}\text{ s}$) として，Verlet 法により数ステップ原子を動かし，各ステップにおける座標を求めよ．このとき，原子は xy 平面上を動くものとする．
- (4) 以上の計算を行う Fortran プログラムを書け．また，(1) の原子配置を適当に変更し，原子 3 個がある程度の距離を保持して振動する初期配置を探れ．(近すぎると弾け飛んでしまうし，遠すぎると相互作用せず全く動かない．)
* ボルツマン定数 $k_B = 1.38 \times 10^{-23}\text{ J/K}$ ，Al 原子の質量 $m = 1.67 \times 10^{-27} \times 27.0\text{ kg}$ ．
- (5) 上記のプログラムを 3 次元に拡張せよ．
- (6) Al 単結晶のユニットセル ($a_0 \times a_0 \times a_0$ ，原子 4 個) を x, y, z 方向に $3 \times 3 \times 3$ 配置し，原子 108 個のセルを作れ．これを初期配置として MD 計算をせよ．
- (7) 3 方向に周期境界条件を課し，無限に大きな Al 単結晶の MD 計算を行え．この時，理想格子に原子を配置すると働く力がゼロとなり原子は全く動かないので，有限温度 $T > 0$ に対する初期速度を以下のように与えよ：

$$v_i^\xi = \sqrt{\frac{k_B T}{m}} \cdot \left(\sum_{j=1}^{12} x_j - 6 \right) \quad (\xi = x, y, z) \quad (62)$$

ただし x は $0 \sim 1$ の乱数である．

- (8) 温度 T を式 (12) を用いて計算し，その変動を求めよ．また，ポテンシャルエネルギーと運動エネルギーの和が保存されていることを確認せよ．

参考：乱数を 12 個発生させるプログラム例 . random() の引数が function を呼ぶ度に変わるので、次々に乱数が生成される。(但し i の初期値は 12345 なので、プログラム走らせる度に新たな乱数が作られる訳ではない。)

```
i=12345
do j=1,12
  print *,random(i)
enddo
end

function random(m)
m=48828215*m
if (m.lt.0) m=(m+2147483647)+1
random=dbl(m)/dbl(2147483647)
end
```

10 演習(2)：ナノワイヤの計算

10.1 1 方向の周期境界条件

周期境界条件のルーチンを適宜修正して、1 方向(例えば z 軸方向)のみに周期境界条件をかけるようにすると、その方向にのびるナノワイヤの計算になる。以下は 3 方向周期境界条件のサンプル。

```
npbcx=rc/lx+1; npbcy=rc/ly+1; npbcz=rc/lz+1
rc = 12.0d-10; rc2 = rc**2; del=1.0d-15; del2 = del**2
! Force and energy calculation
fx = 0.0d0; fy = 0.0d0; fz = 0.0d0; epot = 0.0d0
do i=1,n
  do j=1,n
    do ix=-npbcx,npbcx
      do iy=-npbcy,npbcy
        do iz=-npbcz,npbcz
          drx = rx(i) - rx(j) - lx*ix
          dry = ry(i) - ry(j) - ly*iy
          drz = rz(i) - rz(j) - lz*iz
          rr2 = drx**2 + dry**2 + drz**2
          if ((rr2>del2).and.(rr2<rc2)) then
            rr = sqrt(rr2)
            fx(i) = fx(i)-vp(rr)/rr*drx
            fy(i) = fy(i)-vp(rr)/rr*dry
            fz(i) = fz(i)-vp(rr)/rr*drz
            epot(i)=epot(i)+v(rr)/2.0d0
          endif
        enddo
      enddo
    enddo
  enddo
enddo
```

```

        enddo
    enddo
enddo

```

10.2 構造緩和計算

MD 計算において、運動エネルギーを徐々に取り除いていくことで最適な原子配置を探索することができ、これを構造緩和計算と呼ぶ。速度スケールリングによる温度制御を利用して徐々に設定温度を下げていくことで運動エネルギーを減少させていく方法や、ある温度に到達すると運動エネルギーを取り除く（原子の速度をゼロにする）方法などもあるが、より効率の良い方法を紹介する。

```

! 'Whole atoms' damping (should be more efficient than below)
sp=0.0d0
do i=1,n
    sp = sp+fx(i)*vx(i)+fy(i)*vy(i)+fz(i)*vz(i)
enddo
if (sp < 0.0) then
    vx=0.0d0; vy=0.0d0; vz=0.0d0
endif

```

これは、速度ベクトルと力ベクトルの内積を計算し、それが負になったときに運動エネルギーを取り除く方法である。原子が進んでいる方向と逆の方向に力が加かったときにブレーキをかける、というようなイメージである。プログラムに数行書き足すだけで実現でき、しかも効率が非常に良い。

10.3 応力の計算

応力の定義には mechanical な定義

$$\sigma = \frac{P}{A} \quad (63)$$

と thermodynamics による定義

$$\sigma = \frac{1}{V} \frac{\partial E}{\partial \varepsilon} \quad (64)$$

があり、両者はマクロでは一致するが、原子系では厳密には一致しないことがある（詳細はスライドで説明）。式 (64) はピリアル (virial) 応力とも呼ばれ、原子シミュレーションではこちらが用いられることが多い。

引張り解析などを行い、応力-ひずみ線図を描きたい場合には、まずひずみに対するエネルギー変化をプロットし、そこから式 (64) を用いて応力-ひずみ関係に変換すればよい。

10.4 演習

1. 前回の課題では、アルミ単結晶の有限温度下の計算を行った。この状態から上記の構造緩和計算を行い、運動エネルギーが下がっていく様子を観察せよ。
2. 適当な大きさのナノワイヤモデルを作り、軸方向への引張り解析を行ってみよ。引張りひずみの増分 $\Delta\varepsilon$ を加え、その度に緩和計算を行うと良い（緩和が十分に行われたのを確認してから、次のひずみ増分を加える）。

3. 上記のナノワイヤ引張りにおいて、まずエネルギー-ひずみ関係をプロットせよ。次にそのグラフの傾きから、応力-ひずみ関係を求めよ。構造変化が起こるときに、応力の低下が生じるはずである。

ひずみを z 軸方向に与えるとき、セルの z 方向サイズ L_z を増加させるだけでなく、各原子の座標もセルサイズの増大に応じて計算しなおす必要がある。つまり L_z を $1 + \varepsilon$ 倍にするとき、原子の z 座標すなわち r_i^z も $1 + \varepsilon$ 倍にしてやる必要がある。なお、式 (64) で体積は初期値 V_0 のままでよい (公称応力)。

10.5 atomeye 用のファイル出力のルーチン例

(In main program)

```
! Write configuration data
  if (mod(istep,10)==0) then
    call conv_num_char(istep/10,stri)
    call write_cfgs(22,'config'//stri//'.cfg',n,rx,ry,rz,lx,ly,lz)
  endif
```

(Subroutine 1)

! Write standard cfg format file

```
subroutine write_cfgs(ifile,filename,n,rx,ry,rz,lx,ly,lz)
```

```
  integer :: n
  double precision :: rx(n),ry(n),rz(n),lx,ly,lz
  character(*) :: filename
  integer :: ifile
  double precision :: ang = 1.0d-10, scl_lat = 1.0d0
```

```
  open (ifile,FILE=filename)
```

```
  write (ifile,'(a22,i10)') 'Number of particles = ',n
  write (ifile,'(a4,f10.5,a9)') 'A = ',scl_lat,' Angstrom'
  write (ifile,'(a10,f10.5)') 'H0(1,1) = ',lx/ang/scl_lat
  write (ifile,'(a10,f10.5)') 'H0(1,2) = ',0.0
  write (ifile,'(a10,f10.5)') 'H0(1,3) = ',0.0
  write (ifile,'(a10,f10.5)') 'H0(2,1) = ',0.0
  write (ifile,'(a10,f10.5)') 'H0(2,2) = ',ly/ang/scl_lat
  write (ifile,'(a10,f10.5)') 'H0(2,3) = ',0.0
  write (ifile,'(a10,f10.5)') 'H0(3,1) = ',0.0
  write (ifile,'(a10,f10.5)') 'H0(3,2) = ',0.0
  write (ifile,'(a10,f10.5)') 'H0(3,3) = ',lz/ang/scl_lat
```

```
  do i=1,n
```

```
    write (ifile,'(f10.3,1x,a2,1x,6f15.8)') 1.0,'A1',rx(i)/lx,ry(i)/ly,rz(i)/lz,0.,0.,0.
  enddo
```

```
  close (ifile)
```

```
end subroutine write_cfgs
```

```
(Subroutine 1)
```

```
! Convert number to character
```

```
subroutine conv_num_char(l1,str)
```

```
integer :: l
```

```
character (3) :: str
```

```
l=abs(l1)
```

```
ibase=ichar('0')
```

```
iii=int(l/100)
```

```
ii=int((l-iii*100)/10)
```

```
i=int((l-iii*100-ii*10)/1)
```

```
if (l1<0) stop 'conv_num_char2 error'
```

```
str=char(ibase+iii)//char(ibase+ii)//char(ibase+i)
```

```
end subroutine conv_num_char
```