

拡張アンサンブル法

山内 仁喬

2021 年 10 月 31 日

本章では、まずはじめにマルコフ連鎖モンテカルロ法の基礎を述べる。続いて、拡張アンサンブル法について解説する。

1 マルコフ連鎖モンテカルロ法

まずはじめに、マルコフ連鎖モンテカルロ法の基礎について述べる。マルコフ連鎖とは、次の状態 j をとる確率が直前の状態 i のみで決定されるようにして状態を変化させていく過程のことをいう。

ここでは n 個の状態を持つ系を考えることにする。状態 i は重み w_i をもち、確率 $P(i \rightarrow j)$ で状態 j に遷移する。ここで、状態 i から状態 j への確率流 $v(i \rightarrow j)$ を次のように定義する。

$$v(i \rightarrow j) = w_i P(i \rightarrow j) \quad (1)$$

定常状態を実現するには、系は釣り合い条件

$$\sum_{i=1}^n v(i \rightarrow j) = \sum_{i=1}^n v(j \rightarrow i) \quad (2)$$

を満たさなければならない。式 (2) を満たすような確率流 $v(i \rightarrow j)$ にしたがって状態を遷移させることで、必要なアンサンブル (例えば、カノニカルアンサンブルや定温定圧アンサンブル) を得ることができる。

メトロポリス法や熱浴法は、式 (2) の十分条件である詳細釣り合い条件

$$v(i \rightarrow j) = v(j \rightarrow i) \quad (3)$$

を満たすようなアルゴリズムである。メトロポリスの方法では、状態 i から状態 j への確率流と遷移確率は次のように与えられる。

$$v(i \rightarrow j) = \frac{1}{n-1} \min(w_i, w_j), \quad j \neq i \quad (4)$$

$$P(i \rightarrow j) = \frac{v(i \rightarrow j)}{w_i} = \frac{1}{n-1} \min\left(1, \frac{w_j}{w_i}\right), \quad j \neq i \quad (5)$$

ここで、係数 $1/(n-1)$ は状態 i を除く $n-1$ 個の状態の中からランダムに状態 j を選ぶことに由来している。一方で、熱浴法では状態 i から状態 j への確率流と遷移確率は次のように与えられる。

$$v(i \rightarrow j) = \frac{w_i w_j}{\sum_{k=1}^n w_k}, \quad \forall i, j \quad (6)$$

$$P(i \rightarrow j) = \frac{v(i \rightarrow j)}{w_i} = \frac{w_j}{\sum_{k=1}^n w_k}, \quad \forall i, j \quad (7)$$

メトロポリス法と熱浴法の確率流の計算の模式図を図 ??, 図 ?? に示す。図からわかるように、これらの方法では同じ状態に留まる確率流 $v(i \rightarrow i)$ が存在する。 $v(i \rightarrow i)$ が大きいほど状態を遷移させることができない

め、マルコフ連鎖モンテカルロシミュレーションとしては非効率になってしまう。

続いて、諏訪・藤堂の方法について説明する。諏訪・藤堂の方法は詳細釣り合い条件 (3) を破っており、釣り合い条件 (2) のみ満たすアルゴリズムである。諏訪・藤堂のアルゴリズムは図 ?? に示した通り、以下の手続きで説明される。ここでの手続きは、 w_j の容量を持つ箱 j に、もともと箱 i を満たしていた液体 i を流し込むような操作である。状態 i の重み w_i は箱 i の容量、状態 i から状態 j への確率流 $v(i \rightarrow j)$ は箱 i から箱 j に移す液体の量に対応する。

- Step 1 もっとも大きい重みを持つ状態を選ぶ。もし、もっとも大きい重みを持つ状態が 2 つ以上あった場合、そのうちの 1 つを選ぶ。ここでは、状態 1 がもっとも大きい重み w_1 を持つと仮定する。このような仮定をしても、一般性は失われない。
- Step 2 液体 1 を可能な限り箱 2 に移す。箱 2 に移される液体の量は $v(1 \rightarrow 2)$ である。もし、液体 1 がまだ残っているならば、さらに次の箱 3 に可能な限り液体 1 を移す。次の箱へ液体 1 を流し込む操作を、全ての液体 1 が移動されるまで続ける。最後に液体 1 が注がれた箱を k とする。ここでは確率流 $v(1 \rightarrow 2), \dots, v(1 \rightarrow k)$ を得る。
- Step 3 次に液体 2 を、部分的に液体が入っている箱 k に可能な限り注ぐ。もし、箱 k が満杯になったら、続いて箱 $k+1$ に液体 2 を注ぐ。すべての液体 2 が他の箱 $k+1, \dots$, 箱 l に移すまで操作を続ける。ここでは確率流 $v(2 \rightarrow k), \dots, v(2 \rightarrow l)$ を得る。
- Step 4 他の液体についても順次、次の箱に液体を流し込む操作を行う。もし最後の箱 n が一杯になってしまったら、一番初めの箱 1 を残りの液体で埋める。以上のようにしてすべての液体を他の箱に移すことができる。

以上の操作で得られる確率流 $v(i \rightarrow j)$ を式で表すと、

$$v(i \rightarrow j) = \max(0, \min(\Delta_{ij}, w_i + w_j - \Delta_{ij}, w_i, w_j)) \quad (8)$$

となる。ただし、

$$\begin{aligned} \Delta_{ij} &\equiv S_i - S_{j-1} + w_1 \\ S_i &\equiv \sum_{k=1}^i w_k \\ S_0 &\equiv S_n \end{aligned}$$

と定義した。式 (8) を使うことで、遷移確率 $P(i \rightarrow j) = v(i \rightarrow j)/w_i$ を得ることができる。ここで、諏訪・藤堂の方法での遷移確率を議論する。状態 i から状態 i 、つまりリジェクトされるような確率流は

$$v(i \rightarrow i) = \begin{cases} \max(0, 2w_1 - S_n), & i = 1 \\ 0, & i \geq 2 \end{cases} \quad (9)$$

と計算される。この式は、 $w_1 \leq S_n/2$ のとき、リジェクト率は 0 であることを意味している。したがって、諏訪・藤堂の方法はリジェクト率を最小化するため、効率的にシミュレーションを行うことができる。ここで、系の状態の数が 2 つ ($n = 2$) のとき、諏訪・藤堂の方法はメトロポリスの方法と等価な方法になることに注意する。

2 マルチカノニカル法

カノニカルアンサンブルシミュレーションでは、ボルツマン因子 $\exp[-\beta_0 U]$ に基づいたサンプリングを実現する。ポテンシャルエネルギー U の確率分布は、

$$P_{NVT}(U) = n(U) \exp[-\beta_0 U] \quad (10)$$

である。ここで、 $\beta_0 = 1/k_B T_0$ 、 k_B はボルツマン定数、 T_0 はシミュレーションの温度、 $n(U)$ は状態密度である。状態密度は本来、体積 V と粒子数 N にも依存するが、カノニカルシミュレーションでは体積・粒子数ともに一定であるので、省略することができる。状態密度は U の増加に伴って急激に増加する関数であり、一方でボルツマン因子 $\exp[-\beta_0 U]$ は減少する関数のため、確率分布はベル型の分布をしている。このため、特に低温におけるシミュレーションでは限られたポテンシャルエネルギー空間からしかサンプルすることができない。

この問題を解決するためにマルチカノニカル法 [1, 2, 3, 4] が提案されている。マルチカノニカル法では、ポテンシャルに関して平坦な確率分布（一様分布）に基づいたシミュレーションを実行する。平坦なポテンシャル分布 $P_{\text{MUCA}}(U)$ は、状態密度 $n(U)$ と非ボルツマン重み因子 $W_{\text{MUCA}}(U)$ の積でかけられる。

$$\begin{aligned} P_{\text{MUCA}}(U) &= n(U) W_{\text{MUCA}}(U) \\ &= n(U) \exp[-\beta_0 U_{\text{MUCA}}(U; T_0)] = \text{constant}, \end{aligned} \quad (11)$$

ここで、 T_0 は参照温度であり、 β_0 はその逆温度、 U_{MUCA} はマルチカノニカルポテンシャルエネルギーである。マルチカノニカルポテンシャルエネルギーは、元々のポテンシャルエネルギーからのずれ $\delta U(U)$ を用いて

$$U_{\text{MUCA}}(U; T_0) = U + \delta U(U) \quad (12)$$

と書くこともできる。また、式 (11) より、重み因子 $W_{\text{MUCA}}(U)$ は状態密度 $n(U)$ に逆比例することがわかる。この非ボルツマン因子に基づく統計アンサンブルでは、ポテンシャルエネルギー空間上のランダムウォークが実現するため、エネルギー極少状態に留まりにくくなる。そのため、効率的に構造をサンプルすることが可能である。

2.1 マルチカノニカル MC 法

マルチカノニカル MC シミュレーションを実行するには、メトロポリス法の重み因子として $W_{\text{MUCA}}(U)$ を用いる。ポテンシャルエネルギー U を持つ状態 x から、ポテンシャルエネルギー U' を持つ状態 x' への遷移確率は、

$$P(x \rightarrow x') = \frac{1}{W_{\text{MUCA}}(U)} \min[W_{\text{MUCA}}(U), W_{\text{MUCA}}(U')] \quad (13)$$

$$= \min \left[1, \frac{W_{\text{MUCA}}(U')}{W_{\text{MUCA}}(U)} \right] \quad (14)$$

$$= \min [1, \exp \{-\beta_0 (U_{\text{MUCA}}(U'; T_0) - U_{\text{MUCA}}(U; T_0))\}] \quad (15)$$

$$= \min [1, \exp \{-\beta_0 \Delta U_{\text{MUCA}}\}] \quad (16)$$

と計算される。

2.2 マルチカノニカル MD 法

マルチカノニカル分子動力学シミュレーションを実行するには、ハミルトニアンのパテンシャルエネルギーとしてマルチカノニカルポテンシャルエネルギーを使用する。例えば、能勢の熱浴を使用する場合、ハミルトニ

アンは

$$\mathcal{H}_{\text{Nose-MUCA}} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i s^2} + U(\mathbf{r}) + \frac{p_s^2}{2Q} + gk_B T_{\text{eq}} \ln s + \delta U(U(\mathbf{r})) \quad (17)$$

となる。このハミルトニアンから正準方程式により運動方程式を導く。実時間 t での時間発展方程式は、

$$\frac{d\mathbf{r}_i}{dt} = \frac{\mathbf{p}_i}{m_i} \quad (18)$$

$$\frac{d\mathbf{p}_i}{dt} = \left(1 + \frac{\partial \delta U}{\partial U}\right) \mathbf{F}_i - \frac{\dot{s}}{s} \mathbf{p}_i \quad (19)$$

$$\frac{ds}{dt} = s \frac{p_s}{Q} \quad (20)$$

$$\frac{dp_s}{dt} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} - gk_B T_{\text{eq}} \quad (21)$$

である。この運動方程式 (18)–(21) にしたがって時間発展させることで、マルチカノニカル MD シミュレーションを行うことができる。

2.3 再重法: 単ヒストグラム再重法

任意の物理量 A の温度 T における平均値 $\langle A \rangle_T$ を考える。ここで、物理量 A は運動量に依存しないとする。厳密な状態密度 (あるいはマルチカノニカル重み因子) がわかっている場合、温度 T における物理量 $A(U)$ の期待値は

$$\langle A \rangle_T = \frac{\int dU A(U) n(U) e^{-\beta U}}{\int dU n(U) e^{-\beta U}} \quad (22)$$

あるいは状態が離散化されている場合は、

$$\langle A \rangle_T = \frac{\sum_U A(U) n(U) e^{-\beta U}}{\sum_U n(U) e^{-\beta U}} \quad (23)$$

と計算される。

しかし、一般的には系の厳密な状態密度は事前に分からない。マルチカノニカルシミュレーションでは、短いシミュレーションを繰り返し実行することでマルチカノニカル重み因子 W_{MUCA} を前もって決定しておき、その重み因子に基づいてシミュレーションを行う。長いマルチカノニカルシミュレーションを実行した後、単ヒストグラム再重法 [5] によって状態密度 $n(U)$ の最適解を得ることができる:

$$n(U) = \frac{H_{\text{MUCA}}(U)}{W_{\text{MUCA}}(U)} = H_{\text{MUCA}}(U) \exp[\beta_0 U_{\text{MUCA}}(U)] \quad (24)$$

ここで、 $H_{\text{MUCA}}(U)$ はマルチカノニカルシミュレーションから得られたポテンシャルエネルギー U に関するヒストグラムである。

3 焼き戻し法

焼き戻し法では温度を動的変数として扱い、温度に関する一様分布を実現させる。したがってシミュレーション中に座標と運動量だけでなく、温度が変更されていく。

焼き戻し法では、重み関数

$$W_{\text{ST}}(U; T) \equiv \exp\{-\beta U + f(T)\} \quad (25)$$

に比例した確率で各状態を発生させる。ここで、 $f(T)$ は温度の一樣分布が得られるように導入された関数である。つまり、以下の式が成り立つように決定される:

$$\begin{aligned} P_{\text{ST}}(T) &= \int dU n(U) W_{\text{ST}}(U; T) \\ &= \int dU n(U) \exp\{-\beta U + f(T)\} = \text{constant}. \end{aligned} \quad (26)$$

したがって、 $f(T)$ は形式的に

$$f(T) \propto -\ln \left[\int dU n(U) e^{-\beta U} \right] \quad (27)$$

と求められる。 $f(T)$ は無次元化されたヘルムホルツの自由エネルギーに対応する。

実際のシミュレーションでは温度を離散化する。ここでは M 個の温度を使用すると考える。それぞれの温度を $T_1 < T_2 < \dots < T_M$ の順になっているとしても一般性は変わらない。ある温度に対する無次元化されたヘルムホルツの自由エネルギー $f_m \equiv f(T_m)$ を用いると、焼き戻し法の重み因子は、

$$W_{\text{ST}}(U; T_m) \equiv \exp\{-\beta U + f_m\} \quad (28)$$

である。したがって無次元化されたヘルムホルツの自由エネルギーは

$$f(T_m) \propto -\ln \left[\int dU n(U) e^{-\beta_m U} \right] \quad (29)$$

とかける。焼き戻しシミュレーションは以下の手順で実行される。

1. ある温度 T_m を持つ重み因子 $e^{-\beta_0 U}$ に基づいて、モンテカルロあるいは分子動力学シミュレーションを実行する。
2. 状態を固定したまま、温度 T_m をあらたな値 (T_n に更新する。通常、温度交換の受諾率を高めるために、隣の温度 $n = m \pm 1$ に更新することを試す。パラメータの遷移確率は多くの場合メトロポリスの方法を用いて計算される。サンプリング効率を向上させるために熱浴法, Suwa-Todo の方法を用いる方法も提案されている。

4 レプリカ交換法

続いて、カノニカルアンサンブルにおけるレプリカ交換法 [6] を説明する。ここでは N 個の粒子を持つ系を考える。レプリカ交換法では、互いに相互作用をしない、系のコピー (レプリカ) を M 個用意する。各レプリカは、それぞれ異なる温度 T_m ($m = 1, 2, \dots, M$) でのカノニカルアンサンブルを実現している。レプリカ交換法では、温度とレプリカは 1 対 1 に対応している。したがって、レプリカラベル i ($i = 1, \dots, M$) と温度ラベル m ($m = 1, 2, \dots, M$) は置換関係にある。

$$\begin{cases} i = i(m) \equiv f(m) \\ m = m(i) \equiv f^{-1}(i) \end{cases} \quad (30)$$

ここで、 $f(m)$ は温度ラベルからレプリカラベルへの置換関数、 $f^{-1}(i)$ はレプリカラベルから温度ラベルへの置換関数である。レプリカ交換法における状態 X_α を以下のように、温度ラベルとレプリカラベルの組み合わせ

で表す.

$$X_\alpha = [x_1^{[i(1)]}, x_2^{[i(2)]}, \dots, x_M^{[i(M)]}] \quad (31)$$

$$= [x_{m(1)}^{[1]}, x_{m(2)}^{[2]}, \dots, x_{m(M)}^{[M]}] \quad (32)$$

したがって, 状態 X_α は M 個のレプリカによって指定される. ここで, 各レプリカの状態は座標と運動量の組み合わせ

$$x_m^{[i]} \equiv (q^{[i]}, p^{[i]})_m \quad (33)$$

で指定される. 状態 $x_m^{[i]}$ のハミルトニアン \mathcal{H} は系の運動エネルギー K とポテンシャルエネルギー U を用いて

$$\mathcal{H}(x_m^{[i]}) = K(p_m^{[i]}) + U(q_m^{[i]}) \quad (34)$$

とかける. 温度 T_m におけるカノニカルアンサンブルでは, 各状態 $x_m^{[i]}$ の重みはボルツマン因子

$$w_B = \exp[-\beta_m \mathcal{H}(x_m^{[i]})] \quad (35)$$

で計算することができる. ここで, k_B はボルツマン定数, $\beta_m = 1/k_B T_m$ は逆温度である. 各レプリカは相互作用をしないので, 状態 X_α の重み因子は, 各レプリカ i のボルツマン因子の積

$$w_R(X_\alpha) = \prod_{i=1}^M \exp[-\beta_{m(i)} \mathcal{H}(x_{m(i)}^{[i]})] \quad (36)$$

$$= \prod_{m=1}^M \exp[-\beta_m \mathcal{H}(x_m^{[i(m)]})] \quad (37)$$

で与えられる.

レプリカ交換法では, シミュレーションの途中で温度 T_m を持つレプリカ $i = f(m)$ と温度 T_n を持つレプリカ $j = f(n)$ の温度を交換することを考える. つまり, 状態を以下のように変化させる.

$$X_\alpha = [\dots, x_m^{[i]}, \dots, x_n^{[j]}, \dots] \rightarrow X_\beta = [\dots, x_m^{[j']}, \dots, x_n^{[i']}, \dots] \quad (38)$$

いま, 新しい置換関数 f'

$$j' \equiv f'(m) = j \quad (39)$$

$$i' \equiv f'(n) = i \quad (40)$$

を導入した. 2 つのレプリカ間の温度交換の確率流はメトロポリスの方法を用いて計算される. 式 (4) より

$$v(X_\alpha \rightarrow X_\beta) = C \min(w_R(X_\alpha), w_R(X_\beta)) \quad (41)$$

となる. また, 遷移確率は

$$P(X_\alpha \rightarrow X_\beta) = C \min\left(1, \frac{w_R(X_\beta)}{w_R(X_\alpha)}\right) \quad (42)$$

と計算される. ここで, $C = 1/M C_2$ である. 各レプリカにおける各状態の更新がモンテカルル法に基づく場合は, 座標 q (ポテンシャルエネルギー $U(q)$) のみを考慮すればよいが, 分子動力学法に基づいて各状態を更新する場合は運動量 p も考慮する必要がある. レプリカ交換分子動力学法では, 温度を交換するときに, 条件 $\langle K(p_m^{[i]}) \rangle_{T_m} = \frac{3}{2} N k_B T_m$ を満たすように運動量を次のようにスケールする.

$$p_n^{[i']} = \sqrt{\frac{T_n}{T_m}} p_m^{[i]}, \quad p_m^{[j']} = \sqrt{\frac{T_m}{T_n}} p_n^{[j]} \quad (43)$$

したがって、遷移確率を計算する際に運動量エネルギー K は打ち消されるため、重み因子は式 (37) の代わりに

$$w_R(X_\alpha) = \prod_{m=1}^M \exp[-\beta_m U(q_m^{[i(m)]})] \quad (44)$$

を用いればよい。遷移確率は具体的に次のように計算される。

$$P(X_\alpha \rightarrow X_\beta) = C \min(1, \exp(-\Delta)) \quad (45)$$

ここで、 Δ は

$$\Delta = (\beta_m - \beta_n)(U(q^{[j]}) - U(q^{[i]})) \quad (46)$$

で与えられる。

レプリカ交換シミュレーションは、次の 2 つのステップを交互に繰り返すことで実行される。その様子を模式的に示したのが図??である。

Step 1: 各レプリカ独立に、温度 T_m におけるカノニカルアンサンブルに従うモンテカルロあるいは分子動力学シミュレーションを実行する。

Step 2: 適当なステップごとに、隣接した温度 T_m と T_{m+1} に対応するレプリカ対 i, j 間で温度交換を試みる。式 (45) を用いて遷移確率を計算し、遷移確率にしたがってレプリカ間で温度を交換する。

レプリカ交換法における温度の設定は、どの隣合う温度対に対しても、アクセプト率が一定になるように分布させる。経験的に以下のように指数関数的に決定するとうまくいくことが多い。すなわち、 m 番目の温度を

$$T_m = T_1 \left(\frac{T_M}{T_1} \right)^{\frac{m-1}{M-1}} \quad (47)$$

で与える。ここで、 T_1 は最も低い温度、 T_M は最も高い温度である。

5 レプリカ置換法

続いてレプリカ置換法 [7] について説明していく。レプリカ置換法では、 M 個すべてのレプリカ間で温度の値を置換することを考える。つまり、

$$X_\alpha = [x_1^{[i(1)]}, \dots, x_M^{[i(M)]}] \rightarrow X_\beta = [x_1^{[j(1)]}, \dots, x_M^{[j(M)]}] \quad (48)$$

あるいは

$$X_\alpha = [x_{m(1)}^{[1]}, \dots, x_{m(M)}^{[M]}] \rightarrow X_\beta = [x_{n(1)}^{[1]}, \dots, x_{n(M)}^{[M]}] \quad (49)$$

と状態を遷移させる。ここで、 i, j は温度ラベルからレプリカラベルへの置換関数、 m, n はレプリカラベルから温度ラベルへの置換関数である。レプリカ交換法では 2 つのレプリカ間で温度を交換するだけであったが、レプリカ置換法では 2 つ以上のレプリカ間で温度を置換させる。温度とレプリカの組み合わせは、 $M!$ 個だけあるので、状態を表す α や β は 1 から $M!$ の値をとる。例として、3 つのレプリカを置換する場合の状態の候補を表 1 に示す。

レプリカ置換分子動力学法において温度置換が実行される時、各レプリカの運動量を

$$p_{n(i)}^{[i]} = \sqrt{\frac{T_{n(i)}}{T_{m(i)}}} p_{m(i)}^{[i]} \quad (50)$$

とスケールする必要がある。したがってレプリカ置換法において各状態 X_α の重み因子は、式 (44) を用いて計

表 1: レプリカが 3 つの場合のレプリカ置換状態ラベルの割り当ての例

| 状態ラベル α | レプリカ 1 | レプリカ 2 | レプリカ 3 |
|----------------|--------|--------|--------|
| 1 | T_1 | T_2 | T_3 |
| 2 | T_1 | T_3 | T_2 |
| 3 | T_2 | T_1 | T_3 |
| 4 | T_2 | T_3 | T_1 |
| 5 | T_3 | T_1 | T_2 |
| 6 | T_3 | T_2 | T_1 |

算することができる。レプリカ置換法では遷移確率を計算する際に、メトロポリスの方法を用いると遷移確率 $P(X_\alpha \rightarrow X_\beta)$ は非常に小さい値をとってしまう [7]。つまり、リジェクト率が非常に高くなってしまう。そこで、レプリカ置換法では、メトロポリスの方法に変わって諏訪・藤堂の方法を用いて遷移確率を計算する。諏訪・藤堂の方法はリジェクト率を最小化するためレプリカ置換に適したアルゴリズムである。確率流 $v(X_\alpha \rightarrow X_\beta)$ は、式 (8) 中の重み因子 w_i を $w_R(X_\alpha)$ に置き換えることで計算される。つまり、

$$v(X_\alpha \rightarrow X_\beta) = \max(0, \min(\Delta_{\alpha\beta}, w_R(X_\alpha) + w_R(X_\beta) - \Delta_{\alpha\beta}, w_R(X_\alpha), w_R(X_\beta))) \quad (51)$$

で与えられる。ただし、

$$\Delta_{\alpha\beta} \equiv S_\alpha - S_{\beta-1} + w_R(X_1) \quad (52)$$

$$S_\alpha \equiv \sum_{\beta=1}^{\alpha} w_R(X_\beta) \quad (53)$$

$$S_0 \equiv S_{M!} \quad (54)$$

である。より一般的に、 $w_R(X_\gamma)$ が最大の重みを持つとすると、式 (52), (53) は

$$\Delta_{\alpha\beta} \equiv S_\alpha - S_{\beta-1} + w_R(X_\gamma) \quad (55)$$

$$S_\alpha \equiv \begin{cases} \sum_{\beta=\gamma}^{\alpha} w_R(X_\beta) & \text{for } \alpha \geq \gamma \\ \sum_{\beta=\gamma}^{M!} w_R(X_\beta) + \sum_{\beta=1}^{\alpha} w_R(X_\beta) & \text{for } \alpha < \gamma \end{cases} \quad (56)$$

のように修正される。

レプリカ置換シミュレーションは、次の手順で実行することができる。

Step 0: 表 1 のように、温度とレプリカラベルの組み合わせ全てに対して状態ラベル α を割りあてる。

Step 1: 各レプリカに温度を割り当てて、各レプリカ独立に温度 T_m におけるカノニカルアンサンブルに従うモンテカルロあるいは分子動力学シミュレーションを実行する。

Step 2: 適当なステップごとに、諏訪・藤堂の方法を用いることで温度置換を試みる。式 (44) と式 (51) を用いることで、 $\beta = 1, \dots, M!$ に対する遷移確率 $P(X_\alpha \rightarrow X_\beta) = v(X_\alpha \rightarrow X_\beta)/w(X_\alpha)$ を計算することができる。遷移確率 $P(X_\alpha \rightarrow X_\beta)$ にしたがって、状態 X_α から X_β へとレプリカを置換させる。

Step 3: Step 1 と Step 2 を繰り返す。

レプリカ置換シミュレーションの様子を模式的にしたものを図??に示した。レプリカ置換法では 3 つ以上のレプリカ間での入れ替わりや、離れた温度への遷移が発生し、効率的なシミュレーションを実現する。

5.1 置換関数の構築方法

レプリカ置換法を実行するには、置換関数を構築する必要がある。ここでは、置換をどのようにすれば組み立てることができるかを説明する。まず、 M 個の要素を持つ置換 $P_i^{(M)}$ を次のように定義する。上付の添字は要素の数を表す。

$$\begin{aligned} P_i^{(M)} &\equiv [\sigma_i(1), \sigma_i(2), \sigma_i(3), \dots, \sigma_i(M)] \\ &= \begin{pmatrix} 1, & 2, & 3, & \dots, & M \\ \sigma_i(1), & \sigma_i(2), & \sigma_i(3), & \dots, & \sigma_i(M) \end{pmatrix}. \end{aligned} \quad (57)$$

次に、全ての置換の集合を次のように表す。

$$\{P^{(M)}\} \equiv \left\{ \begin{matrix} P_1^{(M)} \\ P_2^{(M)} \\ P_3^{(M)} \\ \vdots \\ P_{M!}^{(M)} \end{matrix} \right\} = \left\{ \begin{matrix} [1, & 2, & 3, & \dots, & M] \\ [2, & 1, & 3, & \dots, & M] \\ [1, & 3, & 2, & \dots, & M] \\ & & \vdots & & \\ [M, & M-1, & M-2, & \dots, & 1] \end{matrix} \right\}, \quad (58)$$

ここで、 $\{P^{(M)}\}$ は $M! \times M$ の行列である。 M 個の要素を持つ置換の集合 $\{P^{(M)}\}$ は、一つ要素の少ない置換の集合 $\{P^{(M-1)}\}$ に基づいて構築することができる。最初のステップは、 $\{P^{(M-1)}\}$ に M 列目の要素として、 M を付け加えることである。すなわち、

$$\begin{aligned} \{P^{(M)}\}_M &\equiv \{[P_i^{(M-1)}, M]: \text{for } i = 1, \dots, (M-1)!\} \\ &= \left\{ \begin{matrix} [1, & 2, & 3, & \dots, & M-1, & M] \\ [2, & 1, & 3, & \dots, & M-1, & M] \\ [1, & 3, & 2, & \dots, & M-1, & M] \\ & & \vdots & & \\ [M-1, & M-2, & M-3, & \dots, & 1, & M] \end{matrix} \right\}, \end{aligned} \quad (59)$$

ここで、 $\{P^{(M)}\}_i$ の中括弧に対する下付きの添字 i は、追加した数字 M が i 列目にいる置換の集合であることを示す。ここで、転置 $\tau_{j,k}$ を考える。この転置は、置換 $P_i^{(M)}$ の j 番目の列と k 番目の列を入れ替える。すなわち、

$$\tau_{j,k} [1, \dots, j, \dots, k, \dots, M] = [1, \dots, k, \dots, j, \dots, M]. \quad (60)$$

となる。次のステップは、 $\{P^{(M)}\}_M$ の M 列目について、要素 M が 1 番目の列に降りてくるまで、隣接する列との転置を順次繰り返していくことである。つまり、

$$\begin{aligned} \{P^{(M)}\}_{M-1} &= \tau_{M-1,M} \{P^{(M)}\}_M, \\ \{P^{(M)}\}_{M-2} &= \tau_{M-2,M-1} \{P^{(M)}\}_{M-1}, \\ &\vdots \\ \{P^{(M)}\}_2 &= \tau_{2,3} \{P^{(M)}\}_3, \\ \{P^{(M)}\}_1 &= \tau_{1,2} \{P^{(M)}\}_2. \end{aligned}$$

最終的に M 個の要素を持つ置換を次のように得ることができる.

$$\{P^{(M)}\} = \left\{ \left\{ P^{(M)} \right\}_i : \text{for } i = M, M-1, \dots, 1 \right\}. \quad (61)$$

5.2 部分置換の構築方法

レプリカ部分置換の構築方法について述べる [8]. M 個の要素を持つ部分置換関数 $S_i^{(M)}$ を

$$\begin{aligned} S_i^{(M)} &\equiv [\sigma_i(1), \sigma_i(2), \sigma_i(3), \dots, \sigma_i(M)] \\ &= \begin{pmatrix} 1, & 2, & 3, & \dots, & M \\ \sigma_i(1), & \sigma_i(2), & \sigma_i(3), & \dots, & \sigma_i(M) \end{pmatrix}, \end{aligned} \quad (62)$$

とかく. ここで,

$$\sigma_i(l) = \begin{cases} l-1, l, \text{ or } l+1 & \text{when } 1 < l < M, \\ 1 \text{ or } 2 & \text{when } l = 1, \\ M-1 \text{ or } M & \text{when } l = M. \end{cases} \quad (63)$$

のように写像を限定した置換の集合を部分置換と定義する. レプリカ部分置換は, 置換関数の式 (57) と似ているが, 写像 $\sigma_i(l)$ に制限課していることが特徴である. 部分置換の集合を

$$\{S^{(M)}\} \equiv \{S_i^{(M)} : \text{for } i = 1, \dots, N_{\{S^{(M)}\}}\}, \quad (64)$$

と定義する. ここで, $N_{\{S^{(M)}\}}$ は $\{S^{(M)}\}$ に含まれる全置換数を表し, $\{S^{(M)}\}$ は $N_{\{S^{(M)}\}} \times M$ の行列である.

M 個の要素を持つ部分置換 $\{S^{(M)}\}$ は, $\{S^{(M-1)}\}$ と $\{S^{(M-2)}\}$ から構築することができる:

$$\{S^{(M)}\} = \left\{ \begin{array}{ll} [S_i^{(M-1)}, M] & : \text{for } i = 1, \dots, N_{\{S^{(M-1)}\}} \\ [S_j^{(M-2)}, M, M-1] & : \text{for } j = 1, \dots, N_{\{S^{(M-2)}\}} \end{array} \right\} \quad (65)$$

ただし,

$$\begin{aligned} \{S^{(1)}\} &= [1], \\ \{S^{(2)}\} &= \left\{ \begin{bmatrix} 1, & 2 \\ 2, & 1 \end{bmatrix} \right\}. \end{aligned}$$

例えば $M = 3$ の場合, 部分置換は次のように構築することができる.

$$\begin{aligned} \{[S_i^{(M-1)}, M] : \text{for } i = 1, \dots, N_{\{S^{(M-1)}\}}\} &= \left\{ \begin{bmatrix} 1, & 2, & 3 \\ 2, & 1, & 3 \end{bmatrix} \right\}, \\ \{[S_j^{(M-2)}, M, M-1] : \text{for } j = 1, \dots, N_{\{S^{(M-2)}\}}\} &= \left\{ \begin{bmatrix} 1, & 3, & 2 \end{bmatrix} \right\}. \end{aligned}$$

同様に, $M = 4$ の場合の部分置換を以下のようにして組み立てることができる.

$$\begin{aligned} \{[S_i^{(M-1)}, M] : \text{for } i = 1, \dots, N_{\{S^{(M-1)}\}}\} &= \left\{ \begin{bmatrix} 1, & 2, & 3, & 4 \\ 2, & 1, & 3, & 4 \\ 1, & 3, & 2, & 4 \end{bmatrix} \right\}, \\ \{[S_j^{(M-2)}, M, M-1] : \text{for } j = 1, \dots, N_{\{S^{(M-2)}\}}\} &= \left\{ \begin{bmatrix} 1, & 2, & 4, & 3 \\ 2, & 1, & 4, & 3 \end{bmatrix} \right\}. \end{aligned}$$

部分置換関数の数 $N_{\{S^{(M)}\}}$ については, 明らかに次の漸化式を満たす.

$$N_{\{S^{(M)}\}} = N_{\{S^{(M-1)}\}} + N_{\{S^{(M-2)}\}} \quad (66)$$

ただし, $N_{\{S^{(1)}\}} = 1$ と $N_{\{S^{(2)}\}} = 2$ である. この漸化式は解くことができ, 次のように M 個の成分を持つ部分

置換の数が求まる.

$$N_{\{S^{(M)}\}} = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2} \right)^{M+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{M+1} \right\}. \quad (67)$$

よく知られているように, これはフィボナッチ数列である. $N_{\{S^{(M)}\}} = 1, 2, 3, 5, 8, 13, 21, \dots$. 部分置換の数 $N_{\{S^{(M)}\}}$ は全置換の数 $M!$ に比べて常に小さい. 例えば, 全置換の数は 8 個の要素を持つときは $8! = 40,320$, 10 個の要素を持つときは $10! = 3,628,800$ である. 一方, それと同じくらいの部分置換の数を持つ要素数は $M = 23$ あるいは $M = 32$ となる ($N_{\{S^{(23)}\}} = 46,368$, $N_{\{S^{(32)}\}} = 3,524,578$).

ここまでは, 部分置換は要素 l が $l-1$ から $l+1$ のみに移る置換のみを考えていたが, より一般的な部分置換を構築することも可能である. 例えば, 隣の値のみの遷移だけでなく, 2 つあるいは 3 つ離れた値への遷移を含む部分置換の集合を構築することもできる. ここで, 部分置換の範囲を表す ϵ を導入し, 要素 l は, (1) $\epsilon+1 \leq M \leq M-\epsilon$ の時, $M-\epsilon$ から $M+\epsilon$ の間の値に遷移, (2) $M < \epsilon+1$ の時, 1 から $M+\epsilon$ の間の値に遷移, (3) $M > M-\epsilon$ の時, $M-\epsilon$ から M の間の値に遷移, するような部分置換を考える.

範囲 ϵ を持つ部分置換 $\{S^{(M)}\}$ は $\{S^{(M-1)}\}$ を基に構築することができる. 最初のステップは, 部分置換 $\{S^{(M-1)}\}$ の M 列目に要素 M を追加することである: $\{S^{(M)}\}_M \equiv \{[S_i^{(M-1)}, M]: \text{for } i = 1, \dots, N_{\{S^{(M-1)}\}}\}$. ここで $\{S^{(M)}\}_i$ の下付きの i は, i 列目に M がある部分置換の集合であることを表している. 次のステップでは, 部分置換 $\{S^{(M)}\}_M$ の M に対して, 隣接する要素との転置を実行する: $\tau_{M-1,M}\{S^{(M-1)}\}_M$. この時, $N_{\{S^{(M)}\}_M}$ 個の置換が生成される. 生成された置換のうち, M 番目の列の要素が $M-\epsilon$ より小さいものを部分置換 $\{S^{(M)}\}_{M-1}$ と定義する. 同様にして, $\{S^{(M)}\}_i$ ($i = M-1, \dots, 2$) に対して M に関する隣接要素との転置試行 (i.e., $\tau_{i-1,i}\{S^{(M)}\}_i$) を繰り返していく. 全ての転置操作の後, M 列目に $M-\epsilon$ より小さい要素を持つ置換を削除し, 残りの置換を部分置換 $\{S^{(M)}\}_{i-1}$ と定義する. このような隣接要素との置換試行と部分置換の選択は, 要素 M が $(M-\epsilon)$ 列目に移動するまで繰り返す. このようにして, 範囲 ϵ を持つ部分置換の集合

$$\{S^{(M)}\} = \left\{ \{S^{(M)}\}_i: \text{for } i = M, M-1, \dots, M-\epsilon \right\}. \quad (68)$$

を得る. $\epsilon = M$ の時の部分置換は, 全置換と同じである.

6 定温定圧アンサンブルにおける拡張アンサンブル法

6.1 定温定圧アンサンブル

定温定圧アンサンブルでは, 粒子数 N , 圧力 P , 温度 T がパラメータとして指定され, 一定に保たれる. つまり, 圧力 P の共役な量である体積 V は, 圧力にしたがって変動する. 有限体積 V の箱に入った N の粒子を考える. この系は, 粒子の座標 $q \equiv \{q_1, \dots, q_N\}$, 運動量 $p \equiv \{p_1, \dots, p_N\}$, 系の体積 V により指定される. 系のポテンシャルエネルギー $U(q, V)$ は q と V の関数として与えられる. 粒子数 N , 圧力 P , 温度 T 一定の系である定温定圧アンサンブルにおいて, 圧力 P がパラメータとして指定され, それに従って系の体積 V が変動する. ここでは簡単のため, 系の一辺の長さが $V^{1/3}$ の立方体の箱の中にあるとする.

定温定圧アンサンブルでは, 温度 T と圧力 P におけるポテンシャルエネルギー E と体積 V の確率分布 $P_{NPT}(E, V; T, P)$ は以下のように与えられる:

$$P_{NPT}(E, V; T, P) = n(E, V) W_{NPT}(E, V; T, P) \quad (69)$$

ここで, $n(E, V)$ は状態密度, $W_{NPT}(E, V; T, P)$ は分布関数であり,

$$W_{NPT}(E, V; T, P) = \exp\{-\beta(E + PV)\} \quad (70)$$

あるいは

$$W_{NPT}(q, V, P) = \exp \left[-\beta \left\{ \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + U(q, V) + PV \right\} \right] \quad (71)$$

と書かれる。立方体の体積でスケールされた座標 $\sigma = V^{-1/3} \mathbf{q}_k$ を導入すると,

$$\begin{aligned} \exp[-\beta\{U(q, V) + PV\}]dq &= \exp[-\beta\{U(\sigma, V) + PV\}]V^N d\sigma \\ &= \exp[-\beta\{U(\sigma, V) + PV - Nk_B T \log V\}]V^N d\sigma. \end{aligned} \quad (72)$$

となる。

6.2 マルチサーマル・マルチバリック法

マルチカノニカル法の拡張としてマルチバーリック・マルチサーマル法が提案されている??。マルチバーリック・マルチサーマル法ではポテンシャルエネルギーと体積で張られる空間における一様分布を実現させるため、ポテンシャルエネルギー空間と体積空間のランダムウォークをする。そのため、広い範囲の温度・圧力における定温定圧アンサンブルを得ることができる。

ここでは通常の定温定圧アンサンブルの分布関数 $W = \exp[-\beta\{U(q, V) + PV\}]$ の代わりに、マルチバーリック・マルチサーマルエンタルピー \mathcal{H}_{MBT} を導入し、重み関数 $W_{\text{MBT}}\{U(q, V), V\} \equiv \exp[-\beta_0 \mathcal{H}_{\text{MBT}}\{U(q, V), V\}]$ を使用する。ここで $\beta_0 = 1/(k_B T_0)$ は参照温度である。 \mathcal{H}_{MBT} はポテンシャルエネルギーと体積についての確率分布が一様になるように与えられる:

$$P_{\text{MBT}}(E, V) = n(E, V)W_{\text{MBT}}(E, V) \quad (73)$$

$$= n(E, V) \exp[-\beta_0 \mathcal{H}_{\text{MBT}}\{U(q, V), V\}] \quad (74)$$

$$= \text{const} \quad (75)$$

また上の式より、定数を除き形式的に

$$\mathcal{H}_{\text{MBT}} = k_B T_0 \ln n(U, V) \quad (76)$$

が得られる。つまり、状態密度を得ることができればマルチバーリック・マルチサーマルエンタルピーは決定される。

6.2.1 モンテカルロ・シミュレーション

モンテカルロ シミュレーションは、エンタルピー \mathcal{H} をマルチバーリック・マルチサーマルエンタルピー \mathcal{H}_{MBT} に置き換えることで実行される。

6.2.2 MD シミュレーション

一方、MD シミュレーションを行う際には運動方程式の \mathcal{H} を \mathcal{H}_{MBT} に置き換えて数値積分を実行すればよい。

6.2.3 重み因子の決定

重み因子は、短いシミュレーションを繰り返し行う逐次計算法や、ワン・ランダウの方法を用いることが多い。十分分布を広げることができたら、重み因子を固定して、長いマルチカノニカルシミュレーションを行う。

6.3 定温定圧アンサンブルにおける焼戻し法

定温定圧アンサンブルにおける焼戻し法では、温度と圧力自体を動変数として扱い、温度と圧力空間における一様分布を実現させる。したがって、シミュレーション中に座標と運動量だけでなく、温度と圧力が変更されていく。

焼戻し法では、重み関数

$$W_{\text{ST}}(U, V; T, P) \equiv \exp\{-\beta(U + PV) + g(T, P)\} \quad (77)$$

に比例した確率で各状態を発生させる。ここで、 $g(T, P)$ は温度と圧力の一様分布が得られるように導入された関数である。つまり、以下の式が成り立つように決定される：

$$P_{\text{ST}} = \int_0^\infty dV \int_V dq W_{\text{PTST}}\{E(q, V), V; T, P\} = \text{const} \quad (78)$$

したがって、 $g(T, P)$ は形式的に

$$g(T, P) = -\ln \left[\int_0^\infty dV \int_V dq \exp[-\beta\{U(q, V) + PV\}] \right] \quad (79)$$

と求められる。これは、定数を除いた無次元化された Gibbs の自由エネルギーである。

実際のシミュレーションでは、温度と圧力を離散化する。温度を M_0 個、圧力を M_1 個使用すると考える。温度は $T_1 < \dots < T_{M_0}$ 、圧力は $P_1 < \dots < P_{M_1}$ の順になっていると仮定しても一般性は変わらない。ある温度・圧力 (T_{m_0}, P_{m_1}) における無次元化された Gibbs の自由エネルギー $g_m = g(T_{m_0}, P_{m_1})$ を用いると重み関数は

$$W_{\text{ST}}(U, V, T_{m_0}, P_{m_1}) = \exp\{-\beta_{m_0}(U + P_{m_1}V) + g_m\} \quad (80)$$

とかけるので、形式的に

$$g(T_{m_0}, P_{m_1}) = -\ln \left[\int_0^\infty dV \int_V dq \exp[-\beta_{m_0}\{U(q, V) + P_{m_1}V\}] \right] \quad (81)$$

定温低圧焼戻しシミュレーションは以下の手順で実行される。

1. ある温度・圧力 (T_{m_0}, P_{m_1}) を持つ重み因子 $e^{-\beta_{m_0}(U + P_{m_1}V)}$ に基づいてモンテカルロあるいは分子動力学シミュレーションを実行する。
2. 状態を固定したまま、温度・圧力 (T_{m_0}, P_{m_1}) をあらたな値 (T_{n_0}, P_{n_1}) に更新する。パラメータの遷移確率は多くの場合メトロポリスの方法を用いて計算される。サンプリング効率を向上させるために熱浴法、Suwa-Todo の方法を用いる方法も提案されている。

参考文献

- [1] Bernd A Berg and Thomas Neuhaus. Multicanonical algorithms for first order phase transitions. *Physics Letters B*, Vol. 267, No. 2, pp. 249–253, sep 1991.
- [2] Bernd A. Berg and Thomas Neuhaus. Multicanonical ensemble: A new approach to simulate first-order phase transitions. *Physical Review Letters*, Vol. 68, No. 1, pp. 9–12, jan 1992.
- [3] Ulrich H.E. Hansmann, Yuko Okamoto, and Frank Eisenmenger. Molecular dynamics, Langevin and hybrid Monte Carlo simulations in a multicanonical ensemble. *Chemical Physics Letters*, Vol. 259,

- No. 3-4, pp. 321–330, sep 1996.
- [4] Nobuyuki Nakajima, Haruki Nakamura, and Akinori Kidera. Multicanonical Ensemble Generated by Molecular Dynamics Simulation for Enhanced Conformational Sampling of Peptides. *The Journal of Physical Chemistry B*, Vol. 101, No. 5, pp. 817–824, jan 1997.
 - [5] Alan M Ferrenberg and Robert H Swendsen. New Monte Carlo technique for studying phase transitions. *Physical Review Letters*, Vol. 61, No. 23, pp. 2635–2638, dec 1988.
 - [6] Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters*, Vol. 314, No. 1-2, pp. 141–151, nov 1999.
 - [7] Satoru G Itoh and Hisashi Okumura. Replica-Permutation Method with the Suwa–Todo Algorithm beyond the Replica-Exchange Method. *Journal of Chemical Theory and Computation*, Vol. 9, No. 1, pp. 570–581, jan 2013.
 - [8] Masataka Yamauchi and Hisashi Okumura. Replica sub-permutation method for molecular dynamics and monte carlo simulations. *Journal of Computational Chemistry*, Vol. 40, No. 31, pp. 2694–2711, dec 2019.