

圧力一定のモンテカルロシミュレーション コントローラーの開発

9531 覚埜 聡一郎
(情報科学科)

平成 26 年 2 月 23 日

概要

熱平衡状態を議論する上で、自由エネルギーを考えると物質の状態が分かる。しかし、trueVASP(第一原理計算ソフト)で自由エネルギーを求めるプログラムは今までなかった。そこで、モンテカルロシミュレーションを用いて、自由エネルギーを求めるプログラムを開発した。開発には、中井が開発した pseudo(似非)VASP を使用した。

pseudoVASP は、tureVASP よりも短い計算時間で trueVASP のような振る舞いを見せる計算プログラムである。しかし、その出力は規格化されていたため、trueVASP との整合性は取れないでいた。そこで、pseudoVASP の改良を行い、trueVASP と同等の出力を行うようにした。またその pseudoVASP のコントローラーとして、NPT 一定の熱平衡状態を達成するモンテカルロシミュレーションプログラムを開発した。

検証の結果、pseudoVASP は trueVASP と同等の結果を出力する動作を正しく示したが、温度があがるにつれて熱膨張による体積の変化と、phonon 計算で求めた体積の変化に差が見られた。

目次

第1章 緒言	3
第2章 背景	4
2.1 モンテカルロ法	4
2.2 熱平衡モンテカルロ法の原理	4
2.3 NPT 一定のモンテカルロシミュレーション	5
2.3.1 NPT-MC のアルゴリズム	5
2.4 ビリアル状態方程式	8
2.4.1 検証	13
2.5 第一原理計算ソフト VASP	13
2.6 似非 VASP	13
2.7 入出力ファイル	14
2.7.1 POSCAR	14
2.7.2 OUTCAR	15
第3章 実装	17
3.1 pseudoVASP の改良	17
3.1.1 体積変化の考慮	17
3.1.2 Lennard Jones ポテンシャルのパラメータフィッティング	18
3.2 force の計算	19
3.3 Virial の計算	20
3.3.1 コード解説	20
3.4 NPT 一定の MC	21
3.4.1 改良したコード解説	22
第4章 結果	23
4.1 Lennard-Jones ポテンシャルのパラメータフィッティング	23

4.1.1	Maple	23
4.1.2	検証	28
4.2	force の VASP との比較	29
4.2.1	コード解説	30
4.2.2	検証	30
4.3	EV カーブ	30
4.4	NPT モンテカルロシミュレーションによる計算結果	30
4.5	熱膨張	33
第 5 章	議論	37
第 6 章	総括	39

第1章 緒言

近年の第一原理計算ソフトの操作性並びに計算速度の飛躍的な発展によって、現実の材料開発に活用できるエネルギー計算が可能となってきた。材料開発における欠陥エネルギーの計算はほぼ確立されており、界面、表面および添加元素の影響は信頼できる精度で計算されている。

材料の動作や製造プロセスにおいては有限温度での自由エネルギーが必要となる。第一原理計算は原理的に基底状態の計算であるため、有限温度の計算には、調和振動子近似に基づいた phonon 計算のパッケージがいくつか開発、提供されている。しかし、半導体や相変態温度近傍での振る舞いには、非調和な影響を取り込むことが不可欠であるが、そのような計算パッケージは提供されていない。

このような計算には、有限温度での分子動力学シミュレーションが一般的に取られるが、残念ながら自由エネルギーの絶対値を取り出すことは出来ない。一方、モンテカルロシミュレーション (MC) には、自由エネルギーを求める Frenkel 法が存在する。Frenkel 法では、基準状態として Einstein 結晶を取り、そこから連続的にポテンシャルを推移させて積分を実行し、現実系の自由エネルギーを求めるという方法が取られる。このためには、通常の体積一定のモンテカルロシミュレーション (NVT-MC) ではなく、圧力一定のモンテカルロシミュレーション (NPT-MC) が必要となる。

そこで本研究では、第一原理計算ソフト VASP を用いた NPT 一定のモンテカルロシミュレーションのプログラムの開発を目的とする。現実の VASP では膨大な計算時間がかかるため、中井の開発した pseudo(似非)VASP を用いて、モンテカルロシミュレーションのプログラム開発を行う。pseudoVASP で体積変化を取り扱えるように改良する。また、VASP の結果と比較するために、現実の Al の物性を再現するようにポテンシャルをフィッティングし直す。つぎに、西谷研で開発してきた NVT 一定のモンテカルロシミュレーションを改良して NPT 一定のモンテカルロシミュレーションプログラムを開発する。

第2章 背景

本章では、本研究で用いる、モンテカルロシミュレーション、第一原理計算及び第一原理計算ソフト VASP (Viena ab-initio Simulation Package), 圧力を求めるビリアル状態方程式, また pseudo(似非)VASP についての基本となる情報を背景としてまとめて記す。

2.1 モンテカルロ法

モンテカルロ法とは、現在では、乱数を用いたシミュレーションや、数値計算を行う手法の総称であるとされているが、元来は、確率的な要因を含まない決定論的解析問題を解くのに乱数を用いる手法として考案されたものである。理論上は解析的に解くことが可能であるが、解を得ることが困難な場合や計算量が多くなりすぎる場合などにおいて、それらの問題を確率モデルに置き換えて計算を行う。しかし本研究においては、確率的に起こる現象をシミュレートするものであるため、前者の「確率的問題」に分類され、各事象の持つ確率分布に従って乱数を繰り返し発生させ、擬似的に現象を再起させることでサンプルを多量に取得し、統計処理を行う事で、現象の推移・経過を追跡したり、その推定値を得ることが出来る手法である。

2.2 熱平衡モンテカルロ法の原理

モンテカルロ法は前述したとおり、乱数により形の粒子を微視的状态を作成していく手法である。この微視的状态の作成は、分子動力学法の状態での軌跡に相当するものである。ある微視的状态の出現する確率は、対象としている統計集団の確率密度に従うものでなければならないが、現在、モンテカルロシミュレーションに際して圧倒的に広く用いられている Metropolis の方法は、あらかじめ確率密度 (分配関数) を知る必要がないように工夫された方法である。実際問題、確率密度は前もってわからないのが普通である。モンテカルロ法は調べようとする体系の取り方によって、小正準モンテカルロ法、正準モンテカルロ法、大正準モンテカルロ法、その他、に分類されるが、その基本的な概念は同一で

ある．西谷 [1] にどのようにして正準集団の平均が得られるかが，Markov 過程と遷移確率，エルゴード性，詳細釣り合いの条件を用いて説明されている．モンテカルロシミュレーションの重要な変数として採択率がある．採択率は図 2.1 に示したとおり，Energy がマイナスの時は全て採択し，プラスの時には， $\exp(E/k_B T)$ に従って採択率が減少するように取る．このようなエネルギー変化に従って系を次々と生成していくことによって，正準集団に近い状態が高い確率で生成し，系が熱平衡状態になる．採択率は温度を下げる と下がり，平衡に達するまでのシミュレーション時間も長くなる．採択率を上げたいときは原子移動の最大値となる刻み幅を小さくする．それにより採択率は増加するが，熱平衡に達するまでの時間も増加する．従って，全体的なシミュレーション時間の短縮にはこれらのパラメータをうまく選択する必要がある．

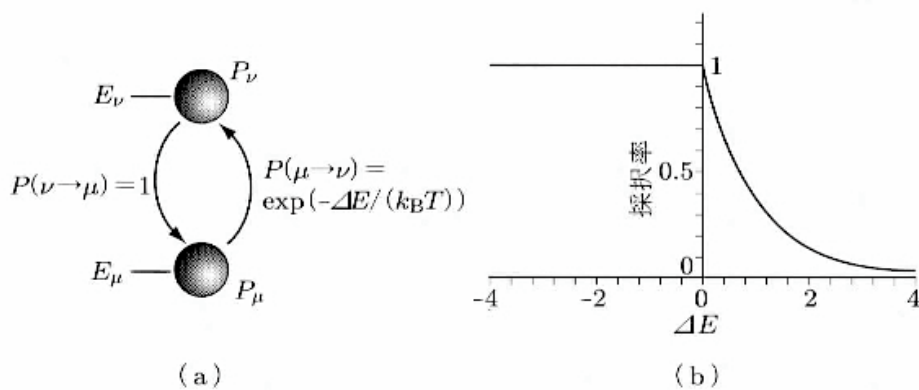


図 2.1: (a) 状態の遷移を表す模式図． (b) 遷移の採択率．

2.3 NPT 一定のモンテカルロシミュレーション

まず初めに，NPT 一定のモンテカルロシミュレーションのアルゴリズムについて記述する．そして，シミュレーションを行うに際してのプログラムの説明を行う．

2.3.1 NPT-MC のアルゴリズム

NPT 一定のモンテカルロ法は，圧力 P を一定にして，体積 V を変化させながら熱平衡状態のシミュレーションを行うアルゴリズムである．NPT-MC を行う際のアルゴリズムを下記に示す．

1. 初期体積と初期状態を与え，その状態を状態 a_i とする．
2. ポテンシャルエネルギー U_i を計算する．
3. 0～1 の範囲に分布する一様乱数列から乱数 R_1 を取り出し，体積を V_i^* から V_j^* に変化させる．すなわち， $V_j^* = V_i^* + (2R_1 - 1) \delta V_{max}$ ．
4. 体積 V_i^* の状態に対し，選び出した一つの粒子を乱数を用いて移動させることにより，状態 a_j をつくる．
5. ポテンシャルエネルギー U_j を求める．
6. エンタルピーを次式にしたがって求める．

$$\Delta H' = PV_0(V_j^* - V_i^*) + U_j - U_i - NkT \ln(V_j^*/V_i^*) \quad (2.1)$$

7. もし， $\Delta H' \leq 0$ ならば，状態 j をマルコフ連鎖を構成する一つの状態とみなして，ステップ3から繰り返す．
8. 上記の乱数列からさらに乱数 R_2 を取り出し，
 - (a) $\exp(-\Delta H'/kT) > R_2$ のとき，状態 j をマルコフ連鎖を構成する一つの状態とみなして，ステップ3から繰り返す．
 - (b) $\exp(-\Delta H'/kT) \leq R_2$ のとき，体積を変化させる前の状態，すなわち，体積が V_i で状態 j で，粒子の位置が s_i をマルコフ連鎖を構成する一つの状態とみなして，ステップ3から繰り返す．

以上の流れを図2.2のフローチャートにて示した．上述のアルゴリズムの圧力 P はビリアル状態方程式から導くことが出来る．

a_i : 原子配置

k_B : ボルツマン定数

$$\Delta H = PV_0(V_j^* - V_i^*) + U_j - U_i - Nk_B T \ln(V_j^*/V_i^*)$$

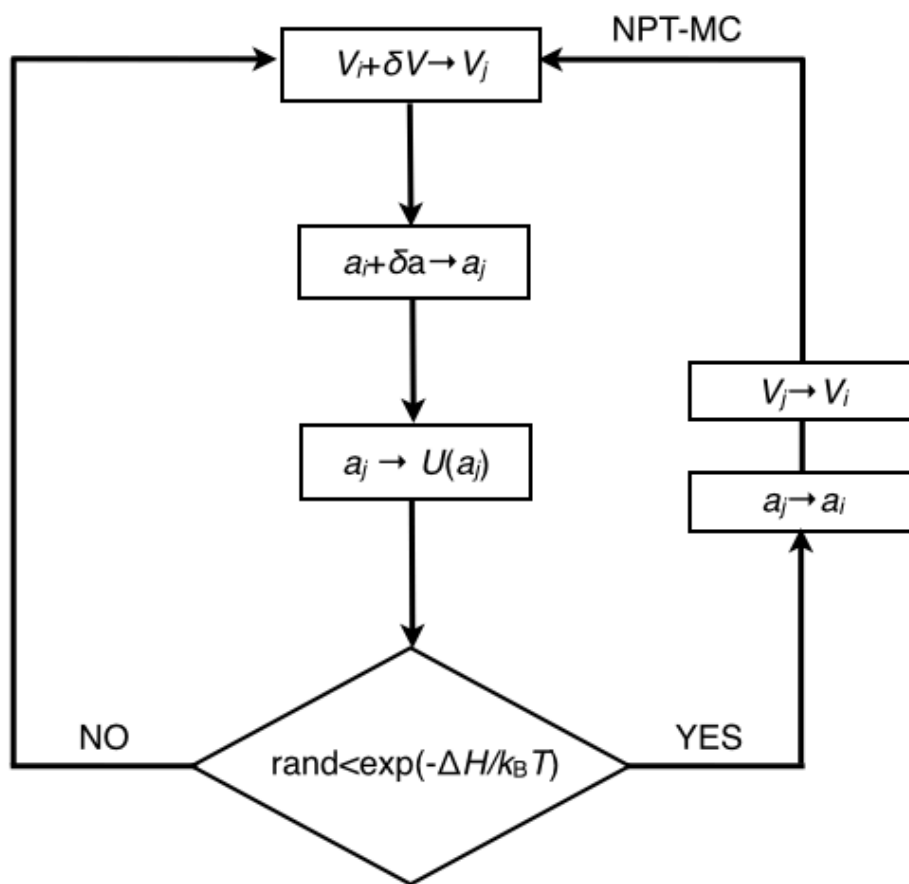


図 2.2: 圧力一定モンテカルロシミュレーションのフローチャート.

2.4 ビリアル状態方程式

ビリアル状態方程式 (virial equation of state) は通常、粒子間および粒子との容器壁との相互作用を考慮することで導出できるが、分子シミュレーションの場合、周期的境界条件を用いるため、容器壁は存在しない。したがって、ここでは通常とは異なる別の方法でこの方程式を、神山新一著、「モンテカルロ・シミュレーション」に従って導出を示す [2]。いま、粒子数が N 、体積が V (一辺が L の立方体、つまり $V = L^3$) なる熱力学的平衡状態にある系を考える。圧力は検査面に垂直に作用する単位面積当たりの力として定

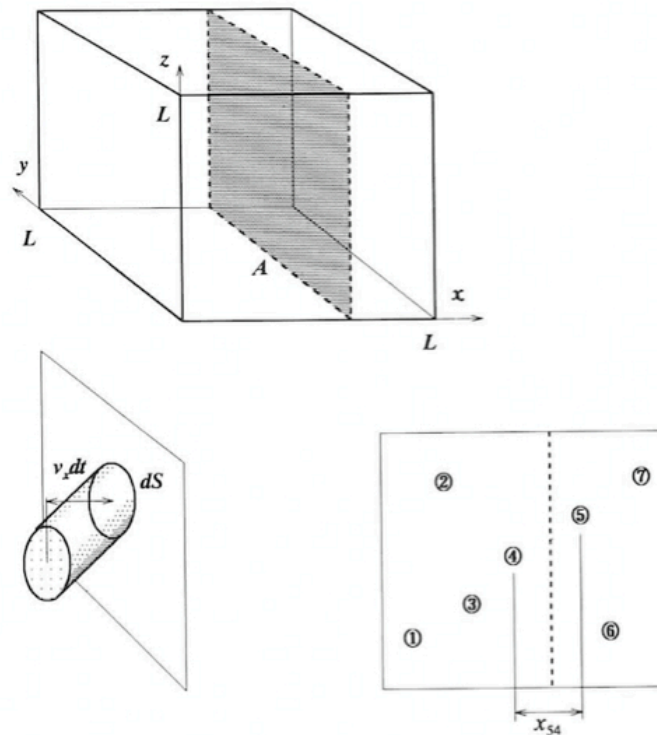


図 2.3: ビリアル状態方程式導出のための検査面。

義される。図 2.3 の場合、検査面の右側部を固体のように考えて、検査面の左側の粒子によって引き起こされる、固体の左側面を圧縮する単位面積当たりの力、と考えればわかりやすい。したがって、図 2.3 の場合、圧力は x 方向の向きを正の方向として通常は定義される。圧力は運動量輸送および粒子間力に起因して発生する。まず最初に、運動量輸送に圧力 P_K を導出し、それから粒子間力による圧力 P_U を導出する。検査面の左側にいる粒子 i に着目し、この粒子が微小時間後に検査面を横切り、右側の領域の粒子と衝突して、また検査面を横切って左側の領域に戻ったとする。この場合の運動量の変化を微小時間で除せば、粒子 i が検査面に作用した力がわかる。しかしながら、系が平衡状態にある

ならば、わざわざ個々の粒子の運動を問題にする必要はなく、統計的に処理することが出来る。すなわち、ある微小時間に検査面に流入する粒子の運動量と流出する運動量を別々に評価することで、検査面に作用する力が計算できる。図 2.3 に示すように、粒子が速度 $v = (v_x, v_y, v_z)$ を有して時間 dt の間に微小検査面 dS を横切ろうとすると、粒子がそのような速度を有する確率は、マクスウェル分布の速度分布 f

$$f(v_i) = \left(\frac{m}{2\pi kT} \right)^{3/2} \exp \left\{ -\frac{m}{2kT} (v_{ix}^2 + v_{iy}^2 + v_{iz}^2) \right\} \quad (2.2)$$

を開いて $f(v_i)dv$ で表すことができ、また、そのような速度を有して微小検査面を横切る粒子数は、平均的に考えれば、図 2.3 の傾斜円柱の体積 $v_x dt dS$ と粒子一個の占める体積 V/N の比で表せる。したがって、速度 v を有して単位面積単位時間あたりに、(左側から) 検査面に流入する運動量の x 方向成分 p_{in}^x は、

$$p_{\text{in}}^x = \left\{ \frac{v_x dt dS}{V/N} f(v) dv m v_x \right\} / (dt dS) = m \frac{N}{V} v_x^2 f(v) dv \quad (2.3)$$

と表せる。この式を速度に関して積分すれば、

$$\begin{aligned} \sigma_{\text{in}}^x &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_0^{\infty} p_{\text{in}}^x dv_x dv_y dv_z \\ &= m \frac{N}{V} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_0^{\infty} v_x^2 f(v) dv \end{aligned} \quad (2.4)$$

となり、 σ_{in}^x は単位面積単位時間あたりに流入する x 方向の運動量である。同様に、検査面から左側へ流出する x 方向の運動量 σ_{out}^x は

$$\sigma_{\text{out}}^x = m \frac{N}{V} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^0 v_x^2 f(v) dv \quad (2.5)$$

したがって、 σ_{in}^x と σ_{out}^x のどちらも正になるようにとってあることを注意すれば、圧力 P_K^x は次のように得られる

$$\begin{aligned} P_K^x &= \sigma_{\text{in}}^x + \sigma_{\text{out}}^x \\ &= m \frac{N}{V} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v_x^2 f(v) dv \\ &= m \frac{N}{V} \langle v_x^2 \rangle \end{aligned} \quad (2.6)$$

通常、圧力は P_K^x , P_K^y , P_K^z の算術平均として定義されるので、この平均値を P_K と書けば、

$$P_K = \frac{1}{3} (P_K^x + P_K^y + P_K^z)$$

P_K^x は P_K の x 方向のみの圧力なので、 $PV = 2E/3$ より、

$$\begin{aligned} P_K &= \frac{1}{3} \left(\frac{2E_x}{V} + \frac{2E_y}{V} + \frac{2E_z}{V} \right) \\ &= \frac{2}{3V} (E_x + E_y + E_z) \end{aligned}$$

ここで E_x , E_y , E_z は単位粒子数あたりの運動エネルギーの $E = mv^2/2$ より

$$P_K = \frac{2}{3V} \left\langle \frac{m(v_x^2 + v_y^2 + v_z^2)}{2} \right\rangle$$

E を全粒子数あたりの運動エネルギーに対応させるために N を掛けると、

$$P_K = \frac{2N}{3V} \left\langle \frac{m(v_x^2 + v_y^2 + v_z^2)}{2} \right\rangle$$

ここで、 $m(v_x^2 + v_y^2 + v_z^2)/2 = E$ より、

$$P_K = \frac{2N}{3V} E$$

$2E/3 = kT$ なので、

$$P_K = \frac{N}{V} kT \quad (2.7)$$

ここに、最後の式は、式 (2.2) を用いて直接平均することで得られた。

次に、粒子間力に起因する圧力 P_U を求める。いま、図 2.3 のように検査面が粒子 4 と 5 の間にあるとする。ただし、粒子はその位置の x 座標が小さいほど、若い番号がつくように番号付けしている。2 体力近似を用いれば、検査面をまたいで相互作用する粒子間の力のみが圧力に寄与する。したがって、図 2.3 の場合、粒子 1 と粒子 4 の間で働く力は圧力に寄与しない。粒子 i に作用する粒子 j の力の x 方向成分を f_{ij}^x とすれば、図 2.3 の場合

の検査面に作用する力の x 方向成分は,

$$-\sum_{i=1}^4 \sum_{j=5}^N f_{ij}^x \quad (2.8)$$

となる. この力は $x_4 \sim x_5$ の座標間で一定である. 検査面を任意の位置に設定し, それらの各位置に対する値を算術平均して検査面に作用する単位面積当たりの力の x 方向成分 r^x を算出すれば, 次のようになる.

$$\begin{aligned} r^x = - \left\{ \left(\frac{x_{21}}{L} \sum_{j=2}^N f_{1j}^x \right) + \left(\frac{x_{32}}{L} \sum_{i=1}^2 \sum_{j=3}^N f_{ij}^x \right) + \right. \\ \left. \cdots + \left(\frac{x_{N,N-1}}{L} \sum_{i=1}^{N-1} f_{i,N}^x \right) \right\} / L^2 \end{aligned} \quad (2.9)$$

ここに, $x_{ij} = x_i - x_j$ である. ゆえに, 右辺を整理すれば,

$$\begin{aligned} r^3 V &= - [x_{21} (f_{12}^x + f_{13}^x + \cdots + f_{1N}^x) \\ &\quad + x_{32} (f_{13}^x + f_{14}^x + \cdots + f_{1N}^x + f_{23}^x + f_{24}^x + \cdots + f_{2N}^x) + \cdots \\ &\quad + x_{N,N-1} (f_{1N}^x + f_{2N}^x + \cdots + f_{N-1,N}^x)] \\ &= \{x_{12} f_{12}^x + (x_{12} + x_{23}) f_{13}^x + \cdots \\ &\quad + (x_{12} + x_{23} + \cdots + x_{N-1,N}) f_{1N}^x\} + \{x_{23} f_{23}^x + (x_{23} + x_{34}) f_{24}^x \\ &\quad + \cdots + (x_{23} + x_{34} + \cdots + x_{N-1,N}) f_{2N}^x\} + \cdots \\ &= (x_{12} f_{12}^x + x_{13} f_{13}^x + \cdots + x_{1N} f_{1N}^x) \\ &\quad + (x_{23} f_{23}^x + x_{24} f_{24}^x + \cdots + x_{2N} f_{2N}^x) + \cdots \\ &= \sum_{i=1}^{N-1} \sum_{j=i+1}^N x_{ij} f_{ij}^x = \sum_{i=1}^N \sum_{j=1}^N x_{ij} f_{ij}^x \end{aligned} \quad (2.10)$$

よって, r^x , r^y , r^z の集団平均の算術平均として P^U を求めれば,

$$\begin{aligned} P^U &= \frac{1}{3} \langle r^x + r^y + r^z \rangle \\ &= \frac{1}{3V} \left\langle \sum_i \sum_j (x_{ij} f_{ij}^x + y_{ij} f_{ij}^y + z_{ij} f_{ij}^z) \right\rangle \\ &= \frac{1}{3V} \left\langle \sum_i \sum_j (x_{ij} \cdot f_{ij}) \right\rangle \end{aligned} \quad (2.11)$$

以上より、圧力 P は式 (2.7) と、式 (2.11) の和として次のように表される、

$$P = \frac{N}{V}kT + \frac{1}{3V} \left\langle \sum_i \sum_j (r_{ij} \cdot f_{ij}) \right\rangle \quad (2.12)$$

この式をビリアル状態方程式という。ここに、 $r_{ij} = r_i - r_j$, f_{ij} は粒子 j が粒子 i に及ぼす力であり、右辺第 1 項は粒子の運動による寄与、第 2 項は粒子間に起因する項である。なお、

$$W = \frac{1}{3} \sum_i \sum_j r_{ij} \cdot f_{ij} \quad (2.13)$$

とおけば、この W を内部ビリアル (internal virial) という。この式は、

$$\begin{aligned} \sum_i r_i \cdot f_i &= \sum_i \sum_j r_i \cdot f_{ij} \\ &= \sum_i \sum_j \left(\frac{1}{2} (r_i \cdot f_{ij}) \right) + \left(\frac{1}{2} (r_i \cdot f_{ij}) \right) \\ &= \sum_i \sum_j \left(\frac{1}{2} (r_i \cdot f_{ij}) \right) + \left(\frac{1}{2} (r_j \cdot f_{ji}) \right) \\ &= \frac{1}{2} \sum_i \sum_j (r_i \cdot f_{ij} + r_j \cdot f_{ji}) \\ &= \frac{1}{2} \sum_i \sum_j (r_i \cdot f_{ij} - r_j \cdot f_{ij}) \\ &= \frac{1}{2} \sum_i \sum_j (f_{ij} (r_i - r_j)) \\ &= \frac{1}{2} \sum_i \sum_j r_{ij} \cdot f_{ij} \end{aligned} \quad (2.14)$$

ここで $(i \neq j)$ の関係にあるとき、 $i = (1, 2, 3)$ と $j = (1, 2, 3)$ は以下の 6 通りである。

$$i, j = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)\}$$

また、 $(i < j)$ の関係にある時、 $i = (1, 2, 3)$ と $j = (1, 2, 3)$ は以下の 3 通りである。

$$i, j = \{(1, 2), (1, 3), (2, 3)\}$$

なので, $(i \neq j) = (i < j) \cdot 2$ である. よって,

$$\begin{aligned} &= \frac{1}{2} \sum_i \sum_j r_{ij} \cdot f_{ij} \cdot 2 \\ &= \sum_i \sum_j r_{ij} \cdot f_{ij} \end{aligned} \quad (2.15)$$

を考慮すると, 次式のようにも書ける.

$$W = \frac{1}{3} \sum_i r_i \cdot f_i \quad (2.16)$$

2.4.1 検証

この計算があっているかの簡単な検証法が思い浮かばなかった. しかし, 単純な計算であるため, バグ等はないものと仮定して以降のプログラム開発を進めている.

2.5 第一原理計算ソフト VASP

本研究では pseudoVASP のコントローラーを作成するので, その動作確認のために trueVASP を用いて pseudoVASP との計算結果の比較を行う. trueVASP は固体物質の電子構造からエネルギーを求める第一原理計算を行うソフトウェアである. trueVASP は平面波基底, 擬ポテンシャル法を用いることで, 高精度な計算を比較的高速に進めている. 西谷研究室では, この trueVASP をの計算を駆動するコントローラーを作成している. 構造緩和や分子動力学, モンテカルロシミュレーションなどこれらのコントローラーでは, trueVASP を使った計算で求めたエネルギーやフォースの値から様々な処理を行い, 新たな原子座標を作成する. そして作成した原子座標を trueVASP に入力として与え, 再び計算を行う. 以上を, 定められた計算終了条件を満たすまで何度も繰り返すことで, 解を求めている. また, 本研究における pseudoVASP の出力は, trueVASP の計算結果を参考に調整している.

2.6 似非 VASP

本研究では, 中井の開発した pseudo(似非)VASP 上で計算シミュレーションを行う [3]. しかし, 計算結果が規格化されており, 計算結果が正であるのか確認出来なかったため,

改良を加えて計算結果が trueVASP と同じになるようにした。pseudoVASP は trueVASP を用いて第一原理計算を行うよりも、短い計算時間でエネルギーやフォースを求めることが出来る。この pseudoVASP は以下の流れで計算を行う。

1. POSCAR から原子座標を読み取る。
2. 読み取った原子座標から近接原子を選択して、ネイバーリストの配列に格納。
3. Lennard-Jones ポテンシャルでエネルギー計算。
4. 求めたエネルギーを距離で微分して、force を計算。
5. 結果を OUTCAR 形式で出力。

2.7 入出力ファイル

ここでは、入力に使用するファイル、及び出力されるファイルの詳細について説明する。

2.7.1 POSCAR

モンテカルロシミュレーションのエネルギー計算をする原子座標の入力には、POSCAR という計算モデルに関するファイルを使用する。POSCAR では、モデル構築において、格子ベクトルなどのユニットセルの形状に関する情報や原子の位置を決定している。今回初期座標として使用するのは、Al の 32 原子の POSCAR であり、内容は以下のようになっている。pseudoVASP では、Direct 以下に表示されている原子座標を読み取り、配列に格納している。また NPT モンテカルロシミュレーションにおける体積変化に関しては、New structure の下の whole scale の値を変化させることで、対応させている。

POSCAR

```
New structure
1.0
      8.0827999115      0.0000000000      0.0000000000
      0.0000000000      8.0827999115      0.0000000000
      0.0000000000      0.0000000000      8.0827999115

32
Direct
0.0000000000      0.0000000000      0.0000000000
0.0000000000      0.0000000000      0.5000000000
0.0000000000      0.5000000000      0.0000000000
0.0000000000      0.5000000000      0.5000000000
0.5000000000      0.0000000000      0.0000000000
0.5000000000      0.0000000000      0.5000000000
0.5000000000      0.5000000000      0.0000000000
0.2500000000      0.2500000000      0.0000000000
0.2500000000      0.2500000000      0.5000000000
0.2500000000      0.7500000000      0.0000000000
0.2500000000      0.7500000000      0.5000000000
0.7500000000      0.2500000000      0.0000000000
0.7500000000      0.2500000000      0.5000000000
0.7500000000      0.7500000000      0.0000000000
0.7500000000      0.7500000000      0.5000000000
0.2500000000      0.0000000000      0.2500000000
0.2500000000      0.0000000000      0.7500000000
0.2500000000      0.5000000000      0.2500000000
0.2500000000      0.5000000000      0.7500000000
0.7500000000      0.0000000000      0.2500000000
0.7500000000      0.0000000000      0.7500000000
0.7500000000      0.5000000000      0.2500000000
0.7500000000      0.5000000000      0.7500000000
0.0000000000      0.2500000000      0.2500000000
0.0000000000      0.2500000000      0.7500000000
0.0000000000      0.7500000000      0.2500000000
0.0000000000      0.7500000000      0.7500000000
0.5000000000      0.2500000000      0.2500000000
0.5000000000      0.2500000000      0.7500000000
0.5000000000      0.7500000000      0.2500000000
0.5000000000      0.7500000000      0.7500000000
0.5000000000      0.5000000000      0.5000000000
```

2.7.2 OUTCAR

pseudoVASP では入力された POSCAR を元に Lennard_Jones ポテンシャルでエネルギーを計算し、結果を OUTCAR というファイルに出力する。OUTCAR は計算終了後に作成されるファイルであり、その計算結果が出力される。計算モデルの計全体のエネルギーやフォース、原子座標、計算時間なども記載されている。以下に、計算終了後に作成される OUTCAR の一例を示す。OUTCAR の出力形式は、pseudoVASP の出力形式になぞらえている。1 行目の配列は格子定数を、2 行目は whole scale を、free energy TOTEN とは、

自由エネルギーの値をそれぞれ表している。また POSITION 以下の 32 列 6 行の値は、左 3 行が原子の位置、右 3 行が force の値である。

OUTCAR

```
[[8.0827999115, 0.0, 0.0], [0.0, 8.0827999115, 0.0], [0.0, 0.0, 8.0827999115]]
1.013
```

FREE ENERGIE OF THE ION-ELECTRON (eV)

free energy TOTEN = -118.0412704 eV

POSITION	TOTAL-FORCE (eV/Angst)				
----------	------------------------	--	--	--	--

0.000000	0.000000	0.000000	-0.056793	0.109081	-0.103280
0.000000	0.000000	0.523937	-0.060490	-0.026110	-0.570900
0.000000	0.507359	0.000000	0.022044	-0.058784	0.018769
0.000000	0.491073	0.477916	-0.061396	0.133995	0.263604
0.507250	0.000000	0.000000	-0.150505	0.032717	-0.111518
0.494342	0.000000	0.495263	0.122770	0.021649	-0.003739
0.495709	0.502472	0.000000	-0.044994	0.020065	-0.116572
0.247752	0.247443	0.000000	0.027928	-0.000993	-0.106504
0.242815	0.246856	0.537945	0.097483	0.130161	-0.719598
0.236901	0.761432	0.000000	0.255092	-0.150466	-0.048948
0.269717	0.739694	0.493170	-0.375774	0.088058	-0.014768
0.744622	0.255629	0.000000	0.017415	-0.350753	0.047026
0.740855	0.258602	0.465722	0.137752	-0.222022	0.460504
0.750450	0.762402	0.000000	-0.015727	-0.239779	-0.134494
0.729522	0.725920	0.489611	0.419892	0.283347	0.088825
0.262073	0.000000	0.247031	-0.194299	0.029219	0.114984
0.253919	0.000000	0.744309	0.051956	-0.170204	0.237541
0.247875	0.485716	0.254956	-0.025601	0.194006	-0.097150
0.240549	0.502810	0.760864	0.056168	0.061927	-0.135565
0.743548	0.000000	0.247868	0.067937	-0.024237	0.071760
0.746551	0.000000	0.743869	-0.097595	0.051055	-0.014006
0.747852	0.486178	0.232381	0.038963	0.339730	0.046695
0.760363	0.465459	0.766074	-0.177621	0.583985	-0.421766
0.000000	0.243755	0.235772	0.051543	-0.057313	0.183713
0.000000	0.249349	0.744351	-0.116474	-0.036970	0.223358
0.000000	0.756512	0.240688	-0.028101	-0.057015	0.263150
0.000000	0.761347	0.742925	-0.075043	-0.326099	0.035816
0.487565	0.260945	0.246845	0.052218	-0.227555	0.005472
0.493317	0.253551	0.741759	0.184496	-0.053076	0.171871
0.500751	0.741796	0.240337	-0.050828	0.055643	-0.007411
0.495689	0.739815	0.736050	0.116400	0.078584	0.307599
0.505262	0.498480	0.495456	-0.188814	-0.211844	0.065531

Total CPU time used: 0.00291

第3章 実装

3.1 pseudoVASP の改良

中井が開発した pseudoVASP では NPT 一定の MC に適用するにはいくつか問題点があった。

- 体積変化が取り入れられていない
- Lennard-Jones ポテンシャルが AI に合っていない

それらを改善するために pseudoVASP に改良を加えた。

3.1.1 体積変化の考慮

コード解説

体積変化を考慮するには、POSCAR に含まれる whole scale を読み込んで、相対座標ではなく、絶対座標でエネルギー計算を行う必要がある。以下に改良を加えたコードを示す。

```

def distance(a,b)
  r=0.0
  scale=$lattice[i][i]*$whole_scale
  for i in 0..2 do
    x=a[i]-b[i]
    x=x-(x/scale).round*scale
    r+=x*x
  end
  return sqrt(r)
end

def f_distance(a,b)
  t=[]
  scale=$lattice[i][i]*$whole_scale
  for i in 0..2 do
    x=a[i]-b[i]
    x=x-(x/scale).round*scale
    t << x
  end
  return t
end

```

3.1.2 Lennard Jones ポテンシャルのパラメータフィッティング

コード解説

Lennard Jones ポテンシャルのパラメータをどのように決めたかは、「結果」において具体的な Maple のコードで示す。ここでは、pseudoVASP で改良を加えた点のコードを示し、その解説を行う。

```
def lj_energy()#LJP
  A=18.19007708
  B=89.22765864
  ene=0.0
  nl.each do |j|
    r=distance(@pos,$atom_list[j].pos)
    ene+/-A*(1/r**3)+B*(1/r**5)
  end
  return ene
end
```

改良した点としては、 A , B の値と、 ene の式の r の指数である。これらの値を変え、VASPでの計算結果と同等の計算結果を出力させることが可能となった。

3.2 force の計算

前述のポテンシャルの変更にともなって原子間の力を計算するルーチンに変更を加えた。以下のプログラムでフォースの計算のコードを示す。

```

def lj_force()
  A=18.19007708
  B=89.22765864
  f=[0.0,0.0,0.0]
  nl.each do |j|
    x,y,z=f_distance(@pos,$atom_list[j].pos)
    r=sqrt(x**2+y**2+z**2)
    dedr=((3*A/(r)**4)-(5*B/(r)**6))
    f[0] += -x/r*dedr
    f[1] += -y/r*dedr
    f[2] += -z/r*dedr
  end
  3.times do |i|
    printf("%10.6f",pos[i]/($lattice[i][i]*$whole_scale))
  end
  3.times do |i|
    printf("%10.6f",f[i])
  end
  printf("\n")
  return f
end

```

x, y, z には、関数 `f_distance` を用いて二原子間の距離を x, y, z 成分ごとに求めて格納している。ここで、エネルギーの距離微分を行っているのは、 $dedr = ((3 * A / (r) ** 4) - (5 * B / (r) ** 6))$ の部分である。

3.3 Virial の計算

3.3.1 コード解説

下記のコードは、virial の計算を行う関数である。式 (2.16) より、原子位置 `force` の積の総和の $\frac{1}{3}$ であることが分かるので、関数 `set_r` にて原子位置の配列を `r` へ、関数 `get_force` にて、`force` の配列を `f` へ格納している。また、その積を変数 `a` に足していきながら代入していく。返す値は、前述の通り、 $\frac{1}{3}$ 倍している。

```

                                virial
def virial  #virial の計算
  a=0.0
  for i in 0..$poscar_text[5].to_f - 1 do
    r=set_r()
    f=get_force()
    for p in 0 ..2 do
      a+=r[i][p]*f[i][p]
    end
  end
  return p=a/3
end

```

3.4 NPT 一定の MC

NPT 一定のアルゴリズムは図 2.2 に示した。ここでは、より実装レベルでどのような計算をさせているかを、データフロー図を用いて解説する。

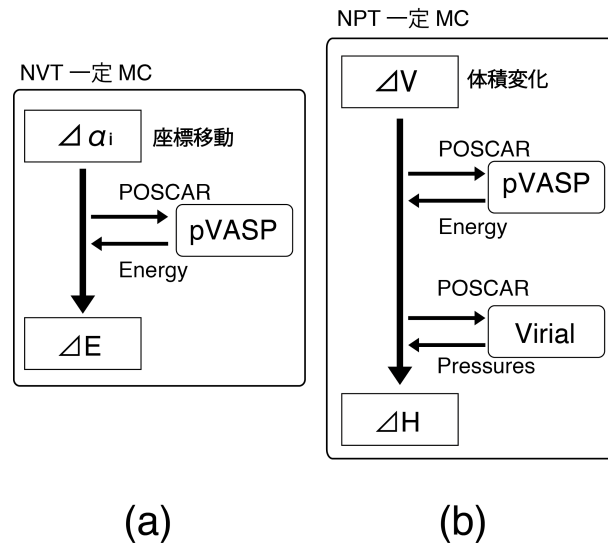


図 3.1: (a)NVT 一定 MC 及び (b)NPT 一定 MC のデータフロー図.

図 3.1(a) は NVT 一定の MC でのエネルギー計算である。原子の移動 (δa_i) に伴って POSCAR を書き換え、pseudoVASP でエネルギー計算を実行している。外部ソフトして

呼び出すのは、今後本物の VASP での切り替えが容易に出来るようにするためである。図 3.1(b) ではこの原子移動に加えて体積変化を考慮する必要がある。その際に、新たに Virial プログラムで圧力計算のもととなる数値を取り出している。これらから、式 (2.1) にしたがって圧力を計算する。

3.4.1 改良したコード解説

PdelV では、採否判断に用いる $P\Delta V$ を計算している。virial() によって外部からプログラムを呼び出し内部圧力を計算している。delH では、式 (2.1) の計算を行っている。

———— NPT-MC 改良 ————

```
def PdelV(volume_before,w_before)
    w_new=virial()
    volume_before
    volume_new = get_volume()
    delV=volume_new-volume_before
    delw=w_new-w_before
    pv=$poscar_text[5].to_f*$Temp+delw
    return pv,w_new,delV
end

def delH(volume_before,energy_before,w_before) #POSCAR から delH を計算
    volume_before
    volume_new=get_volume()
    pv,w_new,delV=PdelV(volume_before,w_before)
    dE=delE(energy_before)
    value = Math.log(volume_new) / Math.log(volume_before)
    delH = pv+dE-($poscar_text[5].to_f*$Temp*value)
    return delH,volume_new,pv,w_new,delV
end
```


第4章 結果

4.1 Lennard_Jonesポテンシャルのパラメータフィッティング

4.1.1 Maple

ばねモデルは、固体の原子レベルの力学的振る舞いを記述する最も適切なモデルである。今回用いる Lennard_Jones ポテンシャルは、ばねモデルを数学的に記述する際によく使用されているポテンシャルであり、二原子間の相互作用エネルギーを求めるのに使用する式である。式 (4.1) にその式を示しておく。

$$\left. \begin{aligned} E_i &= \sum_j \psi(R_{ij}) \\ \psi &= A \left(\frac{1}{R_{ij}} \right)^{12} + B \left(\frac{1}{R_{ij}} \right)^6 \end{aligned} \right\} \quad (4.1)$$

ここで、 ψ は相互作用エネルギー表しており、 R_{ij} は原子 i と原子 j の原子間距離を示している。また、 A, B は適当な定数である。相互作用エネルギー $\psi(R_{ij})$ の距離依存性は図 4.1 のようになる。 $\psi(R_{ij})$ は二原子間の距離 R_{ij} におけるエネルギー、グラフの ϵ は凝集エネルギー (この時の R_{ij} が平衡原子間距離) を表している。結晶の持つエネルギーはすべての原子対についての Lennard_Jones ポテンシャルの和によって与えられる。pseudoVASP での計算結果のグラフが図 4.2 である。改良を加える事で、図 4.1 における ϵ の状態の Energy の値を trueVASP の計算結果である図 4.3 の ϵ の状態の Energy に合わせた。trueVASP での ϵ の状態での計算結果は、 $E=-119.70151874771117158$ 、pseudoVASP では $E=-32.74708792709002618$ となっていた。

Lennard_Jones の式中の A, B の定数に入る適当な数値を Maple を使用して求めた。下記にそのコマンドを示す。

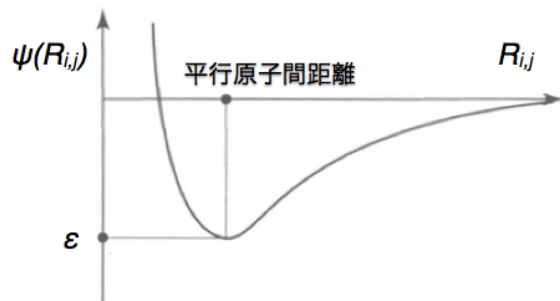


図 4.1: 相互作用エネルギーの距離依存性.

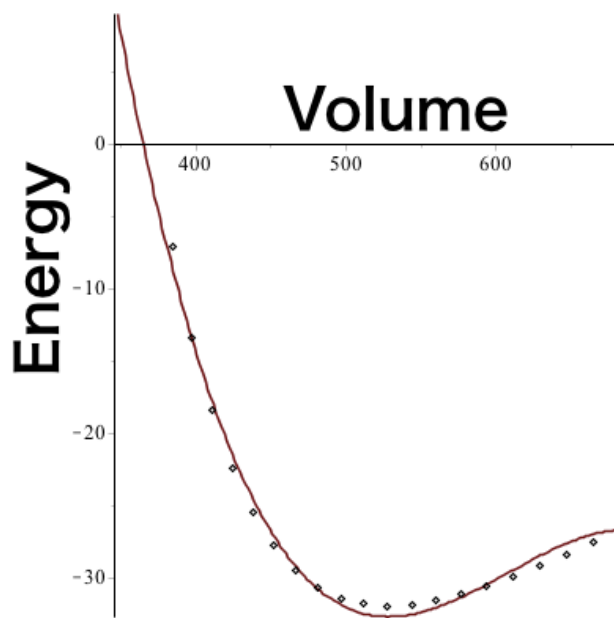


図 4.2: pseudoVASP での計算結果.

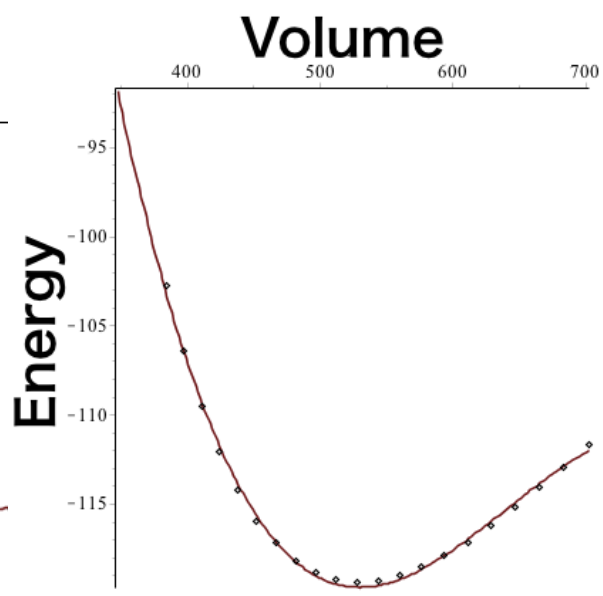


図 4.3: VASP での計算結果.

Maple コマンド

```
Energy_path:="/Users/soichiro/MC/fitting/P_VASPEnergy.txt":  
Energy:=readdata(Energy_path,7);  
Volume_path:="/Users/soichiro/MC/fitting/Volume.txt":  
Volume:=readdata(Volume_path,7);
```

上記式により, VASP で得られた Energy の値を Energy へ, Volume の値を Volume へ配列として格納. また, Energy を単原子あたりの Energy にするため, 原子数 32 で割り, Volume を原子間距離に計算し, data へ格納する.

Maple コマンド

```
for i from 1 to nops(Volume) do  
dataE[i]:=Energy[i,1]/32; #単原子あたりの Energy  
dataV[i:=(Volume[i,1])1/3/2*(sqrt(2)/2); #原子間距離  
data[i]:=[dataV[i],dataE[i]]; #data[] に格納  
end do;
```

その後フィッティングを行い, データ点とフィッティング後の関数を plot して, 確認を行う.

Maple コマンド

```
F2:=unapply(LinearFit([1, x, x2, x3, x4, x5, x6, x7],dataV,dataE,x),x);  
FittingCurve2:=plot(F2(x),x=2.6..3.2);  
pp:=pointplot(data);  
display(pp,FittingCurve2);
```

図 4.4 の最下点, ϵ の状態の x 座標, y 座標を求める.

Maple コマンド

```
v0:=fsolve(diff(F2(x),x)=0,x=2.6..3.2); #VASP 最下点の x 座標 [v0]  
e0:=F2(v0); #VASP 最下点の y 座標 [e0]
```

$$v0 := 2.859295209$$

$$e0 := -3.73513069008374$$

式 (4.1) Lennard_Jones の式における乗数を変えることによって, 曲率を調整するために変

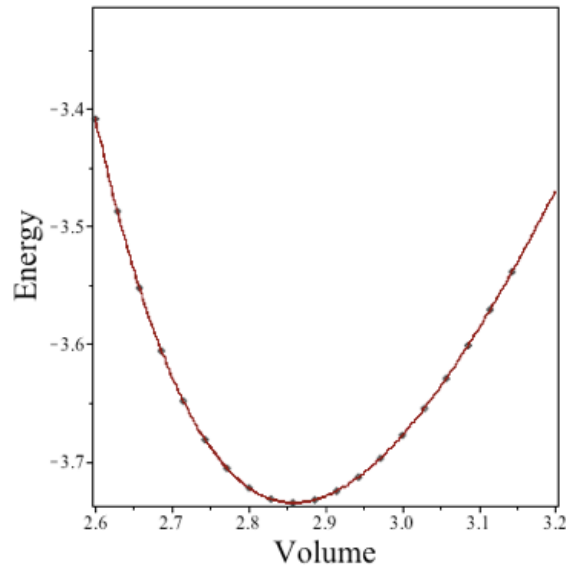


図 4.4: trueVASP での計算結果の plot.

数 n, m を定義, 関数 $f(x)$ に Lennard-Jones の式を代入し, 調整しやすくする.

Maple コマンド

`n:=3:`

`m:=5:`

`f:=unapply(12*(-A*(1/r)**n+B*(1/r)**m),r); #Lennard-Jones → f(x)`

$$f := r \rightarrow -\frac{12 * A}{r^3} + \frac{12 * B}{r^5} \quad (4.2)$$

Maple コマンド

`s1:=solve(diff(f(x),x)=0,x); #極値の x 座標を求める → [s1]`

$$s1 := \frac{1}{3} \frac{\sqrt{15} * \sqrt{A * B}}{A}, -\frac{1}{3} \frac{\sqrt{15} * \sqrt{A * B}}{A}$$

解として正と負の2つの値が出てきたが, 図 4.4 より, 求めたい x 座標は正であるため, 該当する項を変数 $sol1$ へ代入する. 更に, volume である x 座標は trueVASP, pseudoVASP のどちらにおいても同値であるため, 次の事も自明である.

Maple コマンド

```
sol1:=x=s1[1];  
s1[1]=v0;
```

$$\begin{aligned} sol1 := x &= \frac{1}{3} * \sqrt{15} * \frac{\sqrt{A * B}}{A} \\ \frac{1}{3} * \sqrt{15} * \frac{\sqrt{A * B}}{A} &= 2.859295209 \end{aligned} \quad (4.3)$$

Lennard_Jones の式に, ϵ の状態の volume に対応する Energy を VASP のものと対応させるため次式が成り立つ.

Maple コマンド

```
subs(sol1,f(x))=e0;
```

$$-\frac{36}{25} * \frac{A^4 * \sqrt{15}}{(A * B)^{\frac{3}{2}}} + \frac{108}{125} * \frac{B * A^5 * \sqrt{15}}{(A * B)^{\frac{5}{2}}} = -3.73513069008374$$

また, これを B について解くと,

Maple コマンド

```
s2:=solve(subs(sol1,f(x))=e0,B);
```

$$s2 := 0.7092100378 * A^{\frac{5}{3}},$$

$$\begin{aligned} &1.600000000 * 10^9 * (-0.1052682613e - 4 * A^{\frac{1}{3}} + (0.1823299771e - 4 * I) * A^{\frac{1}{3}})^2 * A, \\ &1.600000000 * 10^9 * (-0.1052682613e - 4 * A^{\frac{1}{3}} - (0.1823299771e - 4 * I) * A^{\frac{1}{3}})^2 * A \end{aligned}$$

となる. 2つ目と3つ目の解には虚数が含まれているため1つ目の解を正とし, $sol2$ へ代入する.

Maple コマンド

```
sol2:=B=s2[1];
```

$$sol2 := B = 0.7092100378 * A^{\frac{5}{3}} \quad (4.4)$$

この求めた B を, A, B で表した pseudoVASP における式 (4.3) へ代入することで, 次式が得られる.

Maple コマンド

```
simplify(subs(sol2,s1[1]=v0));
```

$$1.087205928 * \frac{\sqrt{A^{\frac{8}{3}}}}{A} = 2.859295209$$

上記式を, A について解くと,

Maple コマンド

```
s3:=solve(subs(sol2,s1[1]=v0),A);  
sol3:=A:= s3;
```

$$s3 := 18.19037091$$

$$sol3 := A := 18.19037091$$

となり, この値が A の解となる. この値を式 (4.4) に代入することで, B を求める.

Maple コマンド

```
subs(sol3,sol2);
```

$$B = 89.22998054$$

4.1.2 検証

A, B ともに, 値が求まったため引き続き Maple を用いて式 (4.2) へ代入し, 式を求める.

Maple コマンド

```
f3:=unapply(subs(sol2,sol3,f(x)),x);
```

$$f3 := x \rightarrow -\frac{218.2844509}{x^3} + \frac{1070.759767}{x^5}$$

これを, trueVASP の計算結果のデータ点と共に plot する. pseudoVASP は線で, trueVASP は点で plot する.

Maple コマンド

```
d3:=plot(f3(r),r=2.6..3.2);  
display(d3,pp);
```

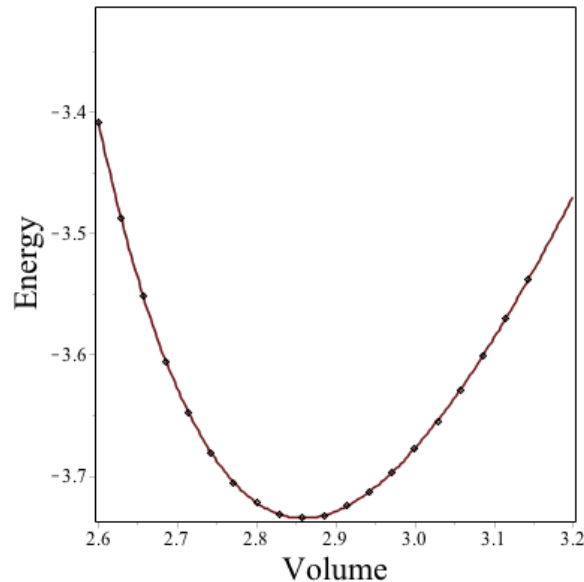


図 4.5: A,B に適当な値を入れた後, trueVASP との値を比較した plot.

図 4.5 の様に trueVASP の計算結果である点と, pseudoVASP によって求まる線が重なっているので, この式にて trueVASP と pseudoVASP の値を合わせることが出来たと言える.

4.2 force の VASP との比較

原子系において原子が安定な位置にあるかどうかを調べるためには, 各原子の持つ force を計算する必要がある. 原子の持つ force が 0 に近ければ近いほど, その原子は安定な位置にあるといえる. force はエネルギーを距離で微分して求める事ができるので, ここでは Lennard-Jones ポテンシャルを用いて求めたエネルギーを原子間の距離で微分する. 前述した Lennard-Jones ポテンシャルの式を改良したため, こちらも改良することで, trueVASP に沿った force の値を求める事ができる. その導出を Mapleで行ったため, 以下に示す. 上記にあるように, force はエネルギーを距離で微分して求める事ができるため, Lennard-Jones ポテンシャルのエネルギーの式を微分するだけで良い.

Maple コマンド

```
Lennard:=unapply((-A*(1/r)**n+B*(1/r)**m),r);  
dedr:=diff(Lennard(r),r);
```

$$Lennard := r \rightarrow -\frac{A}{r^3} + \frac{B}{r^5}$$
$$dedr := 3 * \frac{A}{r^4} - 5 * \frac{B}{r^6}$$

4.2.1 コード解説

4.2.2 検証

以上のコードを実装し、原子を少しずらした同じ POSCAR を用いて trueVASP と pseudoVASP で force の計算を行った。図 4.6 が VASP による計算結果、図 4.7 が pseudoVASP による計算結果である。しかし、そのまま実装したのでは trueVASP と pseudoVASP で force の値の正負が逆になっていたため、各 force の計算の際にマイナスをつけ、trueVASP の結果に沿う様に調整を行った。

4.3 EV カーブ

pseudoVASP と trueVASP の計算結果を横軸に Volume、縦軸に Energy を取って Volume の変化幅を 0.9~1.1 倍の間で plot した。体積が大きくなることで原子間距離も大きくなる。そのため、図 4.1 で示したグラフの様な形になる。図 4.5 は単原子あたりの体積かつ、体積の振れ幅も少ないため、完全一致している様に見えるが、体積が大きくなる時、または小さくなる時にその変化につれて、trueVASP と pseudoVASP とで計算される Energy の値には誤差が生じる。

4.4 NPT モンテカルロシミュレーションによる計算結果

本シミュレーションでは、温度 (T) と試行回数 (n)、各原子の最大移動距離 (d_{max}) をパラメータとして与える。図 4.9 から図 4.11 までのグラフは、それぞれのパラメータが $T = 0.8$, $n = 8000$, $d_{max} = 0.03$ の時のモンテカルロシミュレーションの結果である。またこの時の採択率は 0.429 であった。また、 $t=0.02, 0.06, 0.10$ の時の Energy の推移を

-0.033989	0.000000	0.000000
0.000190	0.000000	0.000000
0.000190	0.000000	0.000000
-0.002094	0.000000	0.000000
0.003089	0.000000	0.000000
-0.000566	0.000000	0.000000
-0.000566	0.000000	0.000000
0.004354	0.005271	0.000000
0.001053	-0.000630	0.000000
0.004354	-0.005271	0.000000
0.001053	0.000630	0.000000
0.004163	-0.005107	0.000000
0.001143	0.000654	0.000000
0.004163	0.005107	0.000000
0.001143	-0.000654	0.000000
0.004354	0.000000	0.005271
0.004354	0.000000	-0.005271
0.001053	0.000000	-0.000630
0.001053	0.000000	0.000630
0.004163	0.000000	-0.005107
0.004163	0.000000	0.005107
0.001143	0.000000	0.000654
0.001143	0.000000	-0.000654
-0.001804	0.000022	0.000022
-0.001804	0.000022	-0.000022
-0.001804	-0.000022	-0.000022
0.000022	-0.000061	-0.000061
0.000022	-0.000061	0.000061
0.000022	0.000061	-0.000061
0.000022	0.000061	0.000061
-0.001984	0.000000	0.000000

図 4.6: VASP での計算結果.

-0.001546	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000238	0.000326	0.000000
0.000000	0.000000	0.000000
0.000238	-0.000326	0.000000
0.000000	0.000000	0.000000
0.000236	-0.000321	0.000000
0.000000	0.000000	0.000000
0.000236	0.000321	0.000000
0.000000	0.000000	0.000000
0.000238	0.000000	0.000326
0.000238	0.000000	-0.000326
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000236	0.000000	-0.000321
0.000236	0.000000	0.000321
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
-0.000087	-0.000001	-0.000001
-0.000087	-0.000001	0.000001
-0.000087	0.000001	-0.000001
-0.000087	0.000001	0.000001
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000

図 4.7: pseudoVASP での計算結果.

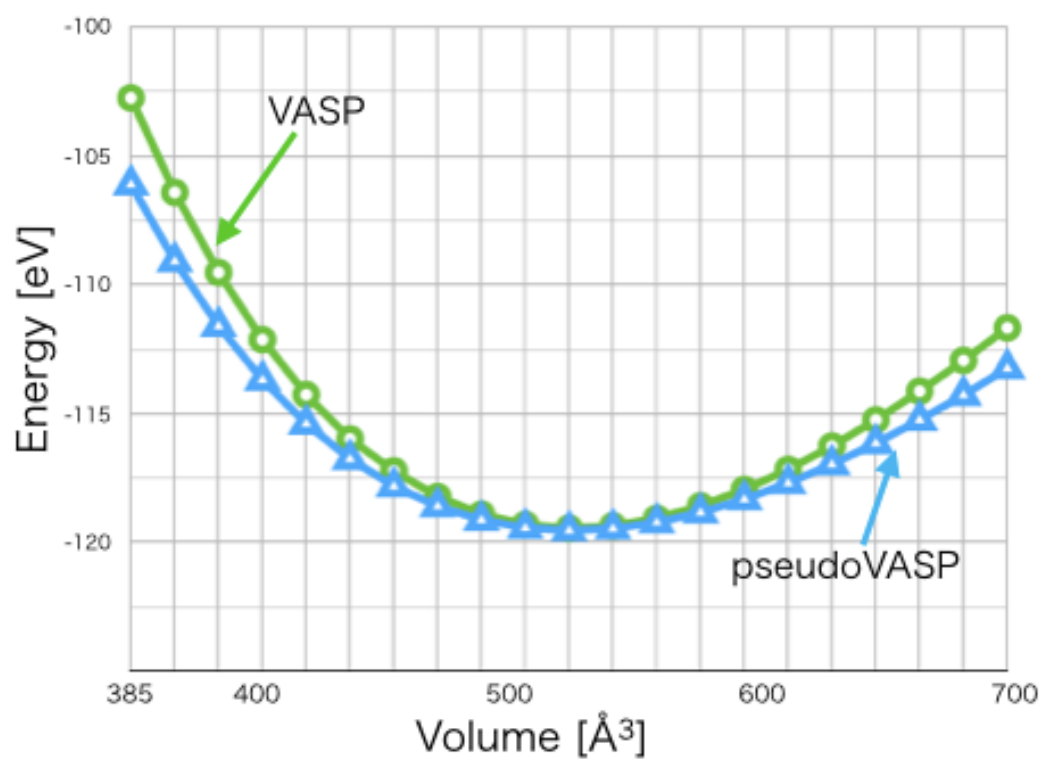


図 4.8: pseudoVASP と trueVASP の EV カーブ.

図 4.12 にまとめて plot した。温度が上がるにつれて、平衡化するエネルギーの値は大きくなっていることが分かる。図 4.10 において、Volume の値が一定間隔幅となっているのは、データに間違いはないが、plot する際の Maple の仕様である。

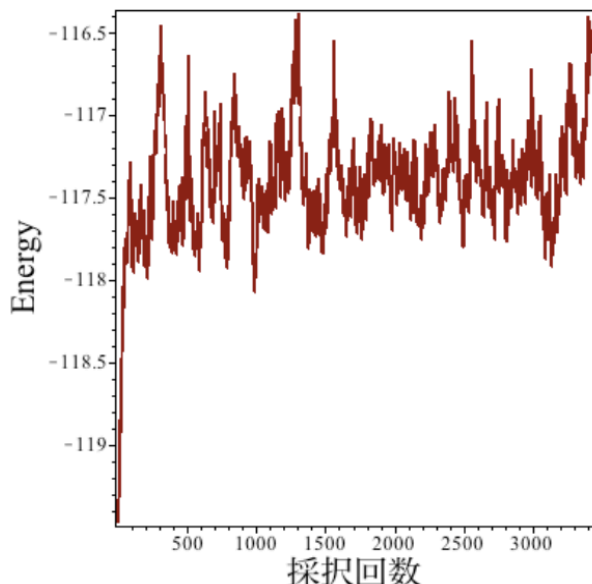


図 4.9: Energy の推移.

4.5 熱膨張

NPT 一定のモンテカルロシミュレーションを温度 $t=0.02 \sim 0.10$ の間でシミュレーションを重ね、エネルギーが平衡化した時の体積の平均を各温度ごとに計算してグラフ化し、温度と体積の関係性を示した。

図 4.13 を見ると、温度が上がるにつれて平衡状態に達した時の体積の平均値が上がっていることが分かる。これは熱膨張によるものであるが、その変動値は図 4.14 で示す phonon 計算による、温度ごとの平衡化した際の体積の平均のグラフと比べると、温度が高くなるにつれて差が出てくる。そこで、温度が $t=0.02, t=0.10$ の時の平衡に達する際の pseudoVASP での体積の振る舞いを図 4.15 に示す。僅かではあるが、その平均値には差があることが分かる。図 4.14 は phonon 計算と一致していないことを意味している。しかし実験との対応を調べる必要がある。実験結果は線膨張係数で示されているので得られた結果を処理して線膨張係数に直した。

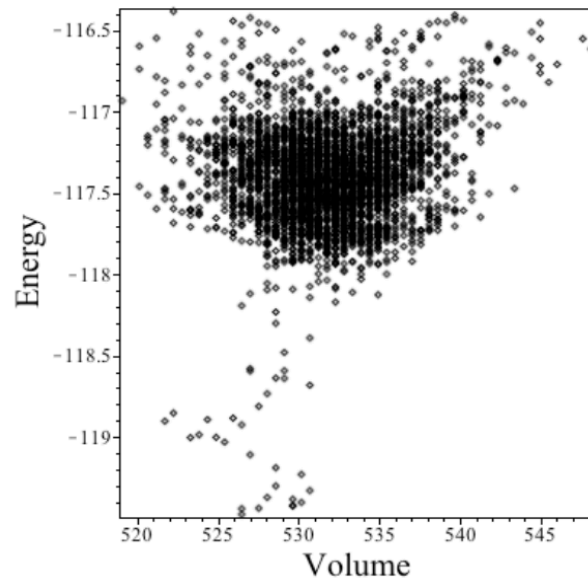


図 4.10: Energy と Volume のポイントプロット.

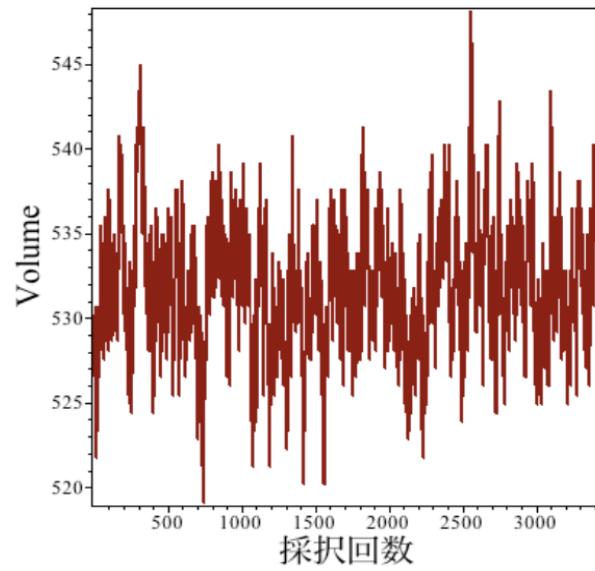


図 4.11: Volume の推移.

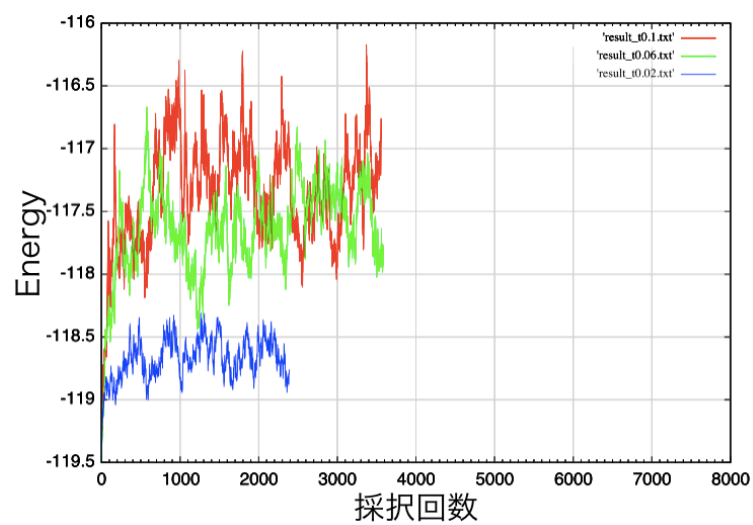


図 4.12: $t=0.02, 0.06, 0.10$ の 3 つの Energy の結果.

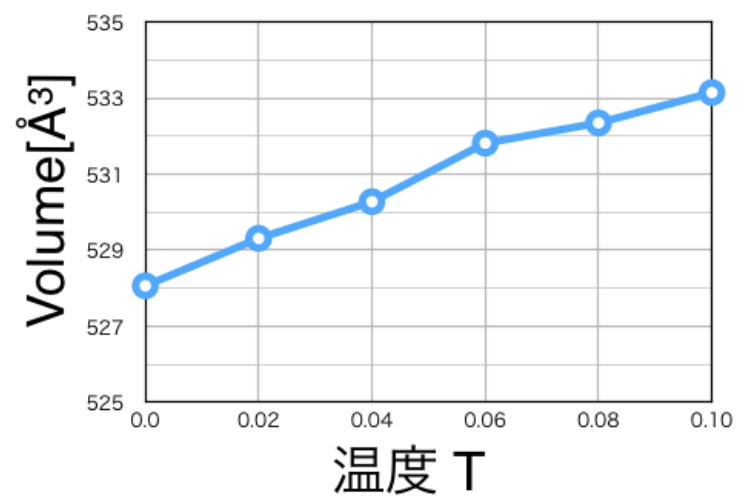


図 4.13: 体積の温度依存性.

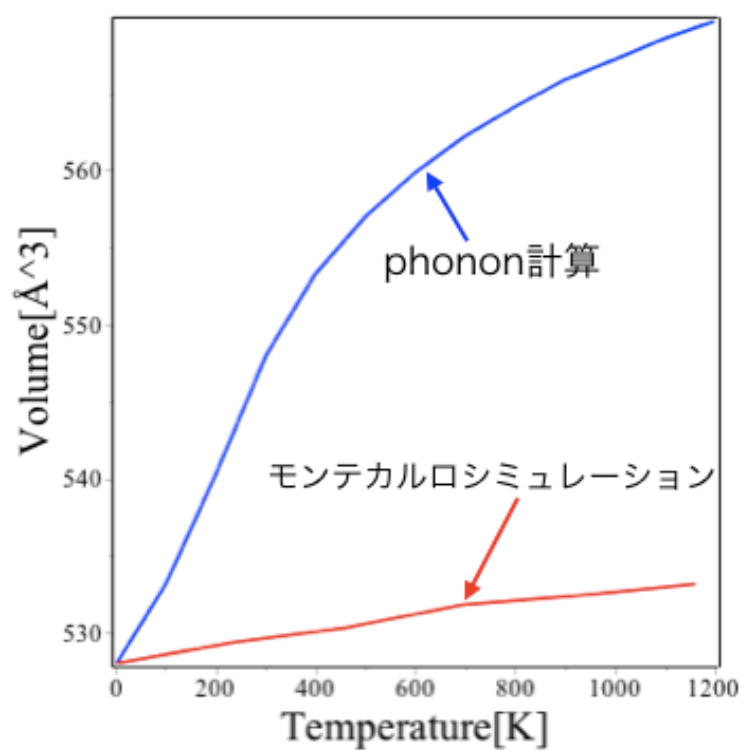


図 4.14: phonon 計算で求めた体積の温度依存性 (青) と図 4.13(赤) を同時に plot したもの.

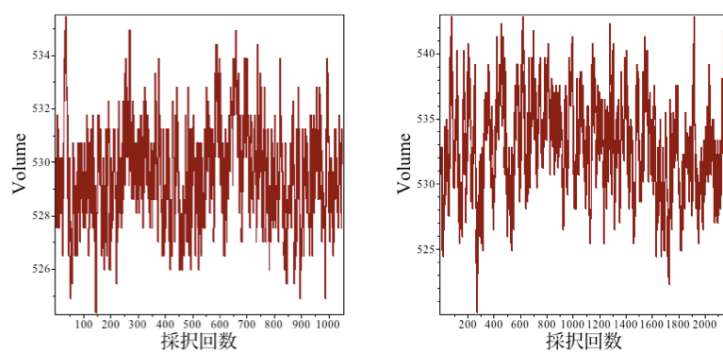


図 4.15: (左) $t=0.02$ の時の Volume. (右) $t=0.10$ の時の Volume.

第5章 議論

以上の結果から pseudoVASP による NPT-MC の動作確認ができた。また、図 5.1 は NPT-MC において、採択された Volume とその時に計算された Energy のグラフである、図 4.10 に二次関数の補助線を入れたものである。このグラフは横軸に Volume, 縦軸に Energy をとっており、図 4.5 と同じく、Energy と Volume の関係を表すグラフである。補助線を入れることでデータ点の集まりの濃淡で、EV カーブが描けていることが分かりやすくなる。このことは、平衡化した際の Volume, Energy は EV カーブの域を出ないということであり、プログラムが正しく動作していることを示している。

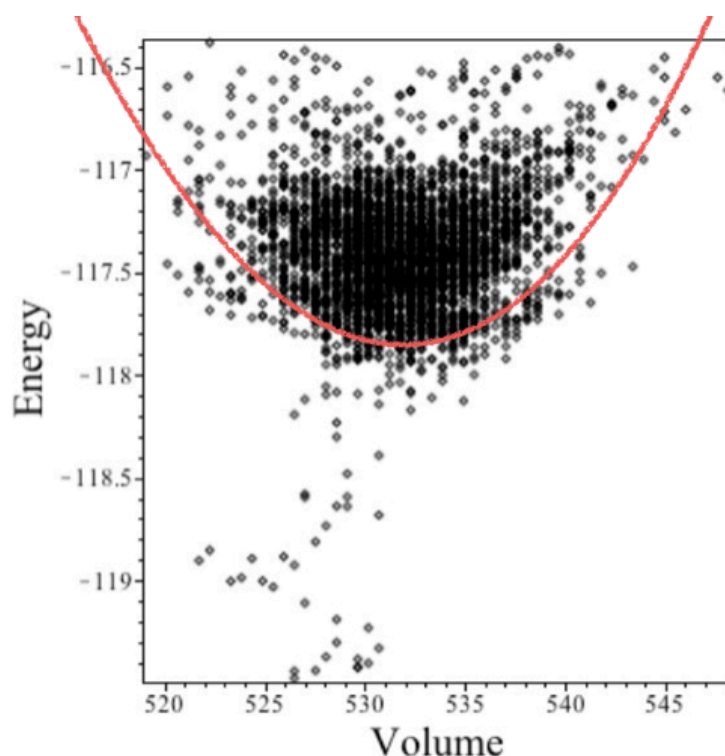


図 5.1: 図 4.10 に補助線を入れたもの。

また、図 5.2 には phonon 計算、実験結果 [4] と NPT-MC の線膨張係数の比較を示した。NPT-MC の計算結果は、phonon の計算及び実験結果からだいぶ離れている。したがっ

て、計算は意図したように進んでいない可能性が高い。今後これらの参照値により近づくように計算プログラムを改良していく必要がある。

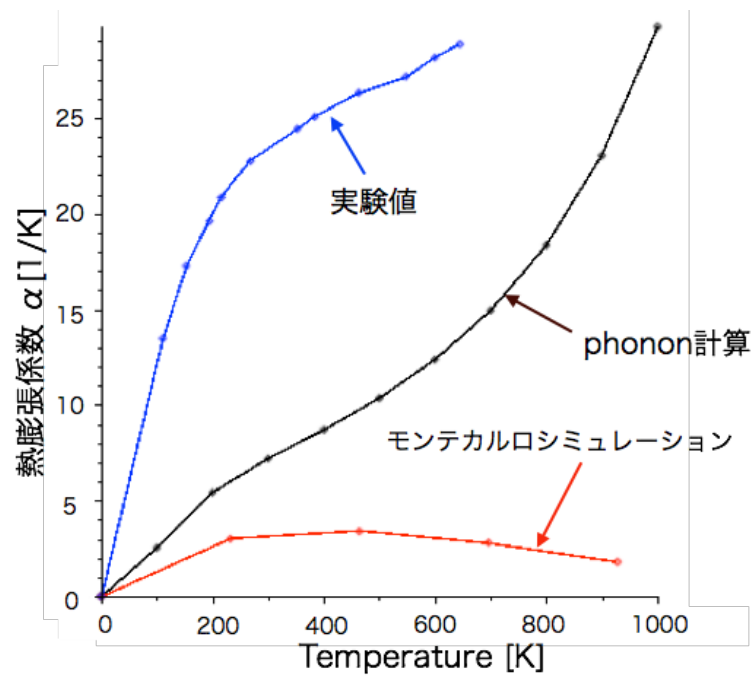


図 5.2: phonon 計算，実験結果 [4] と NPT-MC の線膨張係数の比較.

第6章 総括

本研究では pseudoVASP を用いて、微小な原子移動で一定の熱平衡状態を達成するモンテカルロシミュレーションプログラムを開発、また第一原理計算よりも計算速度の速い原子間ポテンシャルを用いてエネルギー計算を行う pseudoVASP において、trueVASP との出力の値が同等となるように改良を行った。

VASP での Energy の計算結果と、pseudoVASP での計算結果を同等にするため、pseudoVASP 内における Lennard-Jones ポテンシャルの計算プログラムの改良を行い、その動作確認として、Volume と Energy の関係グラフである EV カーブを描き、その重なりを調べることで pseudoVASP での動作に問題がないことがわかった。NPT-MC のプログラムでは、Volume を変化させるため内部圧力を求めるためのビリアル状態方程式を用いてプログラムの開発を進めた。熱膨張によって、温度があがるにつれて体積も増加することを調べるために、phonon 計算を行ってその推移を調べたが体積の変化に誤差が見られたため、今後の課題として、熱膨張の再現の精密化が挙げられる。また、この NPT-MC のプログラムを用いて、trueVASP との連携を行って計算を行うプログラムの開発を行い、精密な計算を行うことも、今後の課題であると考ええる。

謝辞

本研究を遂行するにあたって、終始多大なる御指導、御鞭撻、及び心温かい助言を賜りました西谷滋人教授に対し、深い感謝とともに、心よりの御礼を申し上げます。そして本研究の進行に伴い、同研究室に所属する先輩方、ならびに同輩たち、そして私を支えてくださった家族に、この場を借りて心から感謝致します。本当にありがとうございました。

参考文献

- [1] 西谷滋人著,「固体物理の基礎」(森北出版, 2006)
- [2] 神山新一著,「モンテカルロ・シミュレーション」,(朝倉書店, 1997)
- [3] 中井遙著,「pseudoVASP の開発」関西学院大学理工学部 2012 年卒業論文
- [4] F. C. Nix and D. MacNair,“The Thermal Expansion of Pure Metals: Copper, Gold, Aluminum, Nickel, and Iron”,Phys. Rev. **60** (1941), 597.