1999 年度 修士論文

カーボンナノチューブの面間相互作用に関する研究

電気通信大学 大学院 電気通信学研究科 電子工学専攻

9830055　松尾 竜馬

指導教官 齋藤 理一郎 助教授
木村 忠正 教授

提出日 平成 12 年 2 月 2 日

目次

第 1 章

序論

この章の構成を説明する。 1.1節では本研究の背景を述べる。 1.2節では炭素の同素体であるグラファイト・ダイヤモンド の構造、 1.3節ではフラーレン、 1.4節ではカーボンナノチューブについての構造を述べる。 1.5節で本研究の目的、 1.6節でこの論文の構成を述べる。

1.1 背景

カーボンナノチューブ (CNT:Carbon Nano Tube 以下チューブまたは NT) は、グラファイトシートを継目が無いように筒状に巻いたような構造である。巻き方の違いによりさまざまな直径や螺旋度を持つ CNT が存在する。ナノチューブの構造を一般的に表現する方法として, カイラルベクトルという 2 つの整数値の指数がもちいられる。

太い直径を持つもの NT の中に細い NT、更に細い NT というふうに何重もの入れ子構造の NT は、多層カーボンナノチューブ (MWNT:Multi-Wall Carbon Nanotube) と呼ばれる。対して一層のものを単層カーボンナノチューブ (SWNT:Single-Wall Carbon Nanotube) と呼ばれる。 MWNT の構造のモデルはもう一つある。今紹介した入れ子状の NT は concentric model というが、巻物の様にグラファイトシートが丸まっている scroll model が考えられる。

MWNT の層間にはグラファイトの層を構成するものと同じ分子間相互作用がはたらいてると考えられる。斉藤 (弥) らの電子線 ·X 線回折の実験結果より MWCNT によく見られる層間隔の平均は 3.44Å であり、グラファイト結晶の層間隔よりも広がっている。多層チューブの構造上、結晶グラファイトの安定構造の AB stacking をとることが出来ないことから、 MWCNT の 層の構造は グラファイトでいう 乱層 (Turbostratic) 構造であると考えられる [1]。

坂東らの MWCT の X 線回折の温度依存性の実験より、層の厚い MWNT は scroll model の NT であると考えられる。その根拠は MWNT が 85% に精製された試料 の層間隔の温度依存性がグラファイト結晶の層間隔の温度依存性と一致することである。concentric model であるならば、層間隔の温度依存性は グラファイトの格子定数の依

存性つまり共有結合長の依存性と一致する筈だからである [2]。

ところで MWNT と親戚である、多層フラーレンの層間ポテンシャルの研究が Lu らによって報告されている [3]。結晶 C_{60} は van der Waals 力によって凝集していると考えられるが、その力のポテンシャルを Lennard Jones ポテンシャルで近似して良い結果を得ている [3]。そのポテンシャルは、多層フラーレンの層構造にも良い近似を与え、その層間ポテンシャルを多層フラーレンに適用した結果、層間ポテンシャルは多層フラーレンの球形を球形に維持するように働くことが分った [4]。また Son らはグラファイト表面上の C_{60} の振舞を Lennard Jones 型ポテンシャルをのパラメータを導出して解析した。その計算では、グラファイト表面に 3 角格子状に並べた C_{60} の凝集エネルギーは -22meV/atom であり、C_{60} 同士の距離は中心間距離で 9.896Å である [5]。

Lennard Jones ポテンシャルはフラーレンやグラファイトの分子相互作用を表現するのに成功しているといえる。では多層ナノチューブではどうだろうか。A. Buldum と Jian Ping Lu はグラファイトの表面にナノチューブを置いたときの動力学シミュレーションを Lennerd Jones ポテンシャル [6] をもちいて行なった [7]。NT の動きはスライドとローリングとスピンに分けられる。チューブを回転させて置いた時の相互作用エネルギーは、チューブのカイラル角にユニークな角度で安定な角がある。スライドのエネルギー障壁は、完全なローリングよりも大きくなる。チューブを押した時にスピンしたりスライドする動きが見られる。

多層ナノチューブの場合の研究報告がある。J.-C.Charlier らは、いくつかのアームチェア型の 2 層チューブの場合で LDA による計算を行なった。チューブが 2 層化する時のエネルギーは、グラファイト 2 層の結合エネルギーの 80% であり、チューブのスライド方向のエネルギー障壁が小さいので、短く切られたチューブは楽にチューブ内を出入りできる結果得られた。(5,5)-(10,10) チューブの場合、回転方向に 520 [μeV]、スライド方向に 230[μeV」のエネルギー障壁がある [8]。また、Palser は、tight-binding 法をもちいて (5,5)-(10,10) チューブにおいて原子間結合と分子間斥力を表現し、分子間引力を原子対のポテンシャルで表した計算を行なった。Charlier らの計算値より値は小さいが、回転方向とスライド方向の障壁の大きさはそれぞれに 295μeV、85μeV と見積もられた [9]。

2 層構造 NT の層の相対位置変化は軸回転と、軸方向のズレで与えられる。この 2 つの移動と層間ポテンシャルの変化やチューブペアのカイラリティの依存を理論的に明らかにすることが求められている。

しかし第一原理によるの理論解析が J.-C.Charlier らによって カイラルベクトル (10,10) と (5,5) の 2 層 NT でなされたが、一般のカイラリティのチューブのペアでは、2 つのチューブの軸方向の周期性が一般には非整合であり、またチューブの単位胞が 1000 個を越える場合もあり第一原理の計算や Tight-binding 法 による計算では困難である。そこでフラーレンの原子間 · 分子間相互作用を表現するのに経験的原子ポテンシャルを用いた研究を応用することが考えられる。経験ポテンシャルで炭素原子の共有結合を表現するのに良く用いられるのに Tersoff Potential と分子間のポテン

シャルを表現するのに Lennard Jones 型のポテンシャルが良く用いられている。我々はこの手法を MWCT に応用することを考えた。以下に本研究に必要な炭素材料の構造を述べる。

1.2 グラファイト・ダイヤモンド

天然に見られる炭素の形態は黒鉛 (グラファイト) やダイヤモンドである。黒鉛での炭素は sp^2 結合のネットワークを構成しており、ダイヤモンドでの炭素は sp^3 の結合で結晶を構成している。本研究でもちいる経験ポテンシャルの妥当性を調べるためグラファイトや黒鉛の結晶構造を用いた。その構造を説明する。

1.2.1 グラファイトの構造

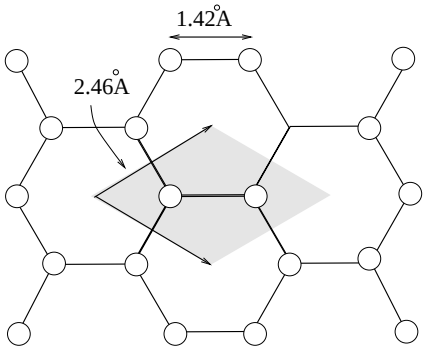

図 1.1 グラフェンの構造 [1]

グラファイトは六角格子構造をもつグラフェン (図 1.1) が積み重なった構造をもつ。グラフェンのユニットセルは灰色の領域で菱形をしている。○が炭素原子で菱形の内部には２つの炭素原子がある。線が sp^2 結合を示していて最近接原子間距離は 1.421Å であり、格子定数は 2.458Å である。

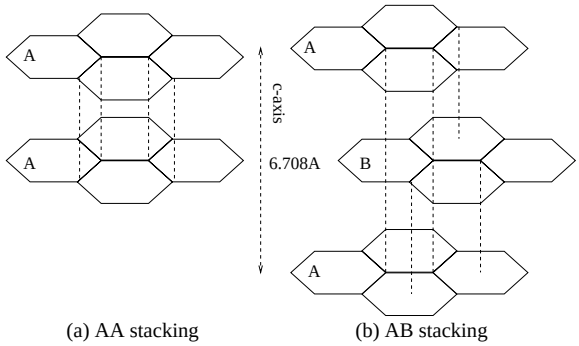

図 1.2 グラファイト積層 [2]

[1] 図 1.1 = m99ryou/gra-unit.eps
[2] 図 1.2 = m99ryou/gra-stack.eps

グラファイトの積層の仕方は、図 1.2(a) のようにの c 軸方向から見て層の相対位置にずれが無い積層を AA stacking、図 1.2(b) の様に 相対位置 A と B が 交互に積み重なるを AB stacking、相対位置 A と B と C が積み重なるのを ABC stacking という。また、層の相対位置にが整合していない構造を乱層 (Turbostratic) 構造という。

単結晶黒鉛に見られる構造は AB stacking であり最も安定である。この構造の層の間隔は 3.354Å であり、乱層構造の場合はそれより間隔が広がること (3.4$\AA$ 程度) が知られている。また AA stacking は安定な構造ではない。 AB stacking グラファイト のユニットセルは 2 層を含むので原子数 4 で c 軸の格子定数は、 6.708Å となる。

1.2.2 ダイヤモンドの構造

ダイヤモンドの構造は、ブラヴェ格子が面心立方格子であり、単位格子に 2 個の原子を含む。格子定数は $a = 3.56\AA$ であり、 [1,1,1] 方向に $(a/4, a/4, a/4)$ だけずれた面心立方格子を重ねあわせた構造である。最近接の原子間距離は 1.544Å である。

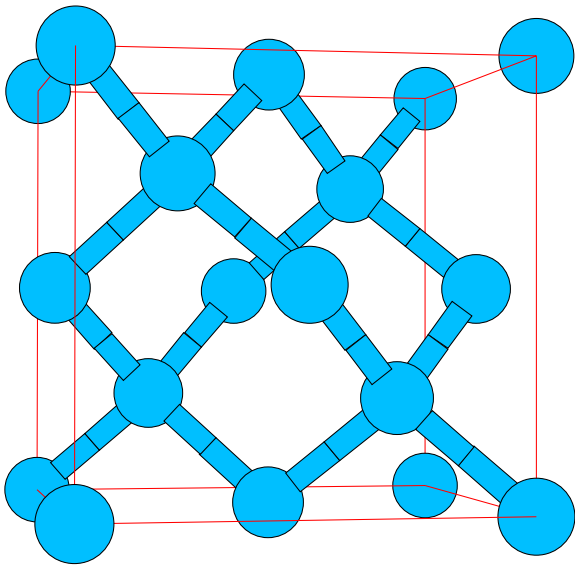

図 1.3 ダイヤモンドの結晶構造 [3]

ダイヤモンドは天然に存在し、世の中で一番硬いものとされている。また宝石としての価値も高い。天然のダイヤモンドは高温高圧の条件で生成されたと考えられる。そのような条件を作り出してダイヤモンドを合成することが実用化されている。

ダイヤモンド結晶は高性能半導体として期待が高まっているが、デバイスに用いることの出来る単結晶を得るのは困難である。しかし、近年、ダイヤモンド薄膜を気相合成する事が可能である事が分かり、多くの研究が行なわれている。

[3]図 1.3 = m99ryou/dia-cell.eps

1.3 フラーレン

フラーレンとは、C_{60} を代表とする炭素ネットワーク状物質である。Kroto や Smally らによって 1985 年に発見された。

1.3.1 フラーレンの構造

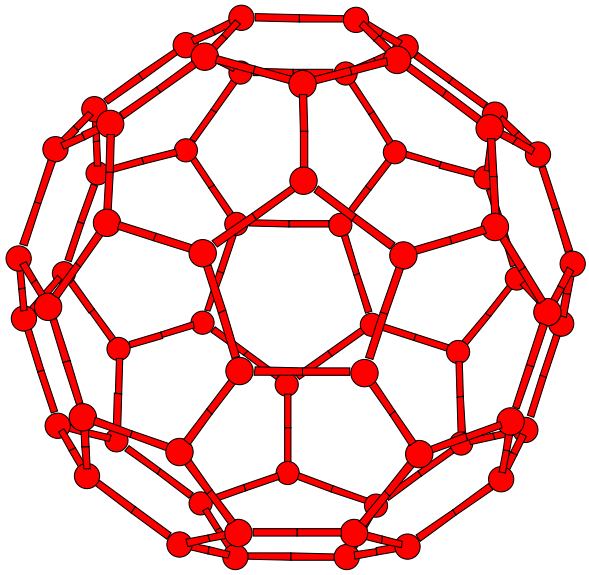

図 1.4 C_{60} の原子模型 [4]

ここでは、C_{60} の構造を紹介する。NMR による測定によると５角形と６角形の接する辺の結合 (5-6 bond) の長さは 1.447Å、６角形同士が接する辺の結合 (6-6 bond) の長さは 1.40Å である。5-6 の結合は 1 重結合的なので single-bond 、6-6 の結合は２重結合的なので double-bond と呼ばれる。この物質の構造を理論計算で表現することができる。構造最適化計算には第一原理による方法や、タイトバインディングによる方法などがあるが、それらの計算による結合長は実験結果と良く一致している。これらは計算量が原子の個数の３乗に比例 (Order N) かそれ以上であるが、岡等は計算量が O(N) になるよう経験ポテンシャルを用いての構造最適で C_{60} を表現することができる事をしめした [10]。

[4]図 1.4 = m99ryou/c60.eps

1.4 カーボンナノチューブ

カーボンナノチューブは飯島 澄男らによって、 1991 年に発見された [11]。カーボンナノチューブは、グラファイトのシートを丸めて筒状にしたような構造をしていて１次元的な構造をしている。この物質で特に注目に値する点は、その構造を決める指数 (カイラル指数) で電子的性質 (金属・半導体) を一意に決められることが理論計算により明らかにされている [12][13][14]。

1.4.1 カーボンナノチューブの構造

カーボンナノチューブの構造は カイラルベクトルと２つの整数値のペアで指定することが出来る [12]。カイラルベクトル (C_h) を指定すると、チューブの直径 (R やカイラル角 θ、チューブの並進ベクトル T、単位格子あたりの原子数 N を計算で求めることが出来る。

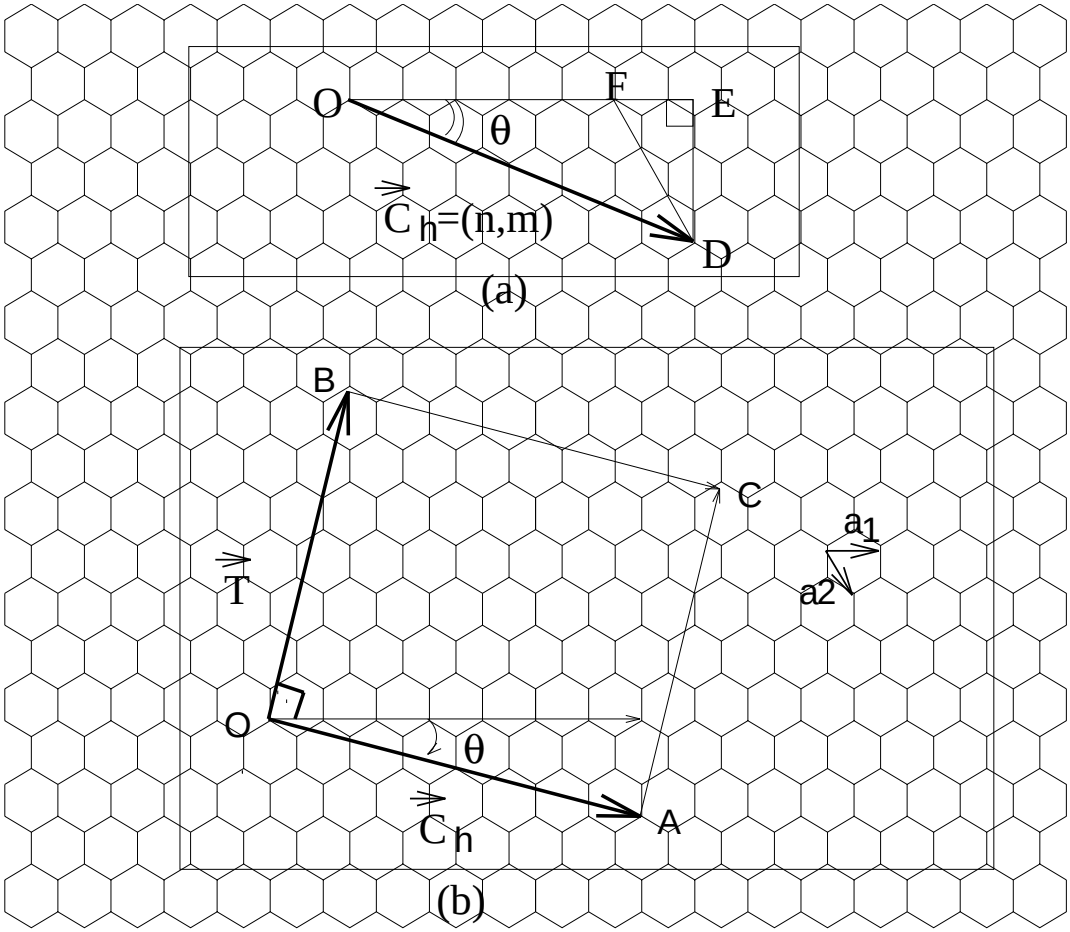

図 1.5 チューブの展開図 (T.Takeya 卒業論文 (1997) より [15])
C_h=(6,2), 半径 R=2.8Å,$|T|$ = Å,N = 104, θ=13.9 度 [5]

[5]図 1.5 = m99ryou/tenkai.eps

カイラルベクトル： $\mathbf{C}_h$

カイラルベクトルにより NT の筒構造を表すことが出来る。カイラルベクトルは 2 つの整数値 (n,m) $(0 \leq |m| \leq n)$ の組で指定でき、グラファイトシートの切りかたを表現している。カイラルベクトル $\mathbf{C_h}$ は グラファイトの基本格子ベクトル $\mathbf{a}_1$,$\mathbf{a}_2$ を用いて表現される。

$$\mathbf{C_h} = n\mathbf{a_1} + m\mathbf{a_2} \equiv (n,m) \qquad (n,m\text{は整数}, 0 \leq |m| \leq n). \tag{1.4.1}$$

直径： $\mathbf{d}_t$

また $\mathbf{C_h}$ によって示された線分がチューブの円周になる。点 O· 点 A を通り 線分 OA に垂直な直線でグラファイトシートを切り、切った線を合わせるように繋げることによりチューブ構造が出来る。炭素原子距離 a_{c-c} が 1.42Å とすると、
a=$|\mathbf{a_1}|$=$|\mathbf{a_2}|$=$\sqrt{3}a_{c-c}$=2.46Å を用いて、チューブの円周は、

$$|\mathbf{C_h}| = L = \sqrt{OE^2 + ED^2} = a\sqrt{n^2 + m^2 + nm}, \tag{1.4.2}$$

で与えられ、またチューブの直径 d_t は $d_t = \frac{L}{\pi}$ で与えられる。

カイラル角： θ

$\mathbf{a_1}$ と $|\mathbf{C_h}|$ のなす角をカイラル角 θ とよぶ、

$$\theta = \tan^{-1}\frac{\sqrt{3}m}{2n+m}. \tag{1.4.3}$$

また $(0 \leq |m| \leq n)$ の条件より、 $|\theta| \leq 30\,^\circ$ が与えられる。
付録 A.1に d_t と θ と C_h を d_t の小さい順に並べた。

チューブの並進ベクトル： $\mathbf{T}$

図 1.5(b) において、 $\mathbf{O}$ から $\mathbf{C_h}$ に垂直な方向ににある $\mathbf{O}$ と最初に等価な格子点を $\mathbf{B}$ とおく。チューブの並進ベクトル $\mathbf{T}$ はベクトル $\mathbf{OB}$ である。 $\mathbf{T}$ は $\mathbf{a_1}$、 $\mathbf{a_2}$ を用いて次式で表される。

$$\mathbf{T} = t_1\mathbf{a_1} + t_2\mathbf{a_2} \equiv (t_1,t_2) \quad (\text{ただし} t_1,t_2\text{は互いに素}) \tag{1.4.4}$$

ここで、 t_1,t_2 は $\mathbf{C_h}$ と $\mathbf{T}$ は垂直なことをもちいて内積の関係 $\mathrm{C}_h \cdot T = 0$ から、、以下のように表される。

$$t_1 = \frac{2m+n}{d_R},\ t_2 = -\frac{2n+m}{d_R}, \tag{1.4.5}$$

ここで d_R は、 (2m+n) と (2n+m) の最大公約数である。

チューブのユニットセルと原子数 : $2N$

チューブのユニットセルは図 1(b) で $\mathbf{C_h}$ と $\mathbf{T}$ からなる長方形 $\mathbf{OABC}$ である。このユニットセル内の六員環の数 N は面積 $|\mathbf{C_h}\times\mathbf{T}|$ を六員環 1 個の面積 $(|\mathbf{a_1}\times\mathbf{a_2}|)$ で割ると、求められ次式のようになる。

$$N = 2\frac{(n^2+m^2+nm)}{d_R} \tag{1.4.6}$$

これよりチューブのユニットセル内の炭素原子の数は、$2N$ となる。

多層ナノチューブ

ナノチューブは内部に空洞を持つので、内部に更に直径の小さいナノチューブが入りそうなことは容易に予想できる。実際、最初に発見されたナノチューブはそのような 多層ナノチューブであった [11]。

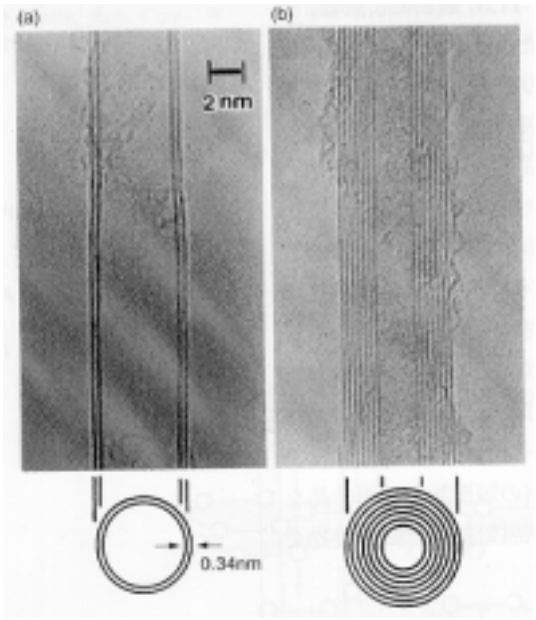

図 1.6 多層ナノチューブの SEM 像
Nature 354(1991) より引用 [6]

多層ナノチューブの層構造は、グラファイトの構造と類似して考えられる。グラファイトの層間は $3.354\mathring{A}$ で AB stack 構造が よくとられるが、その構造からずれがあると、層の間隔は それよりも広がり ($3.4\mathring{A}$ 以上) Turbostratic 構造と呼ばれる。ナノチューブでは、層間の構造が グラファイト結晶と違い、互いの面の位置関係が一致しない為、 Turbostratic 構造であると考えられる。

[6]図 1.6 = m99ryou/mwcnt.eps

1.5 本研究の目的と計算方法の選択

本研究では、２層カーボンナノチューブの内側と外側のカイラリティの組合せの違いによる、安定構造を解析するのが目的である。

多層ナノチューブの面間構造を考えるにあたり、内側とその外側の組合せによる安定構造を考えることは応用上も有用である。面間の相互作用は長距離力であるが、再近接層間の相互作用が重要であり、本研究では2 層構造ナノチューブに問題を絞る。解析には２層カーボンナノチューブの立体構造が必要である。その立体構造を得るには、多くの原子を含む炭素クラスターを構造最適化する必要がある。我々は本研究において、炭素結合のネットワークの構造最適化を行なうプログラム作成した。このプログラムでは原子数が 10000 を超える炭素クラスターの構造最適化が可能である。

構造最適化を行なう手法でクラスター全エネルギーの最小値を与える立体構造を計算する必要がある。しかし系のエネルギーを精密に計算するために第一原理計算の密度汎関数法によることも考えられるが計算時間がかかる。そこで、その代わり経験的原子ポテンシャル関数をもちいる方法を採用した。本研究では炭素の共有結合を Tersoff ポテンシャルで表現し、グラファイトや NT の面間の相互作用を Lennard Jones 型の ポテンシャルでモデルを行なった。

構造最適化の計算プログラムは、計算量が原子の数に比例する $O(N)$ になるよう計算アルゴリズムを最適化され、炭素の共有結合と分子間力の両方を考慮に入れた計算が可能である。このプログラムを応用してナノチューブへのグラファイトパッチ吸着とカイラリティの関係と 2 層チューブの安定構造とカイラリティの関係を調べる。

1.6 本論文の構成

ここでは本論文の構成を述べる。 2章では、本研究でもちいた経験的原子ポテンシャルである Tersoff ポテンシャルと Jennard Jones ポテンシャルの関数形とそのパラメータに関して説明した。それぞれの関数のパラメータは 本研究の対象の系に適合するよう修正が加えられている。つまり Tersoff ポテンシャルはそのままではグラファイトやチューブの原子間距離が実際の系よりも大きな値をとるので結合の長さをどれだけ修正したらいいかを検討した。また Lennard Jones ポテンシャルの定義は原子間距離が無限遠まで定義されてあるが本研究の系で実際に有効なポテンシャルの到達範囲があるり、その到達範囲を設定し、計算時間の短縮を実験した。

3章では、構造最適化の手法を説明した。構造最適化の手法には、共役勾配法や最急勾配法等がある。今回、チューブ構造の構造最適は共役勾配法では不適であることが分り、チューブ構造の最適化に適した最適化手法を述べる。そして最適化法に関連して、数値微分の手法について説明した。系のエネルギーポテンシャルを数値微分をすることにより、ポテンシャル関数の 2 次関数化 (共役勾配法) を行なったり、原子に加わる力を計算する (分子動力学的方法) 必要がありその手法を説明する。その中で計算自体の最適化の手法として最近接原子の探索方法を紹介する。

4章では、ナノチューブへの グラファイト小片の吸着を考える事は、カーボンナノ

チューブの生成過程に関わる事であり、興味深い問題である。今回は、カイラリティの異なる チューブに C_{24} や C_{54} を吸着させて、その安定配置や吸着エネルギーを計算した。

5章では、多層構造のナノチューブの安定構造はどのようなに決まるかを計算で明らかにした。 2層構造のチューブの場合での安定構造を考えることは、理論面での応用上も有効な方法である。今回、さまざまなカイラリティの組での、2層チューブの相互作用のエネルギーや2層のチューブ相対位置とエネルギーポテンシャルの関係を解析した。6章で、本修士論文で得られた主な結論をまとめた。

第 2 章

経験的原子ポテンシャルによる構造最適化

経験的ポテンシャルは、それを表現する関数において実験などから得られた又は実験に適合するように定めたパラメータを用いている。それに対して第一原理による計算ではできるかぎりのもしくは全く別に与えられパラメータを使わない。経験ポテンシャルを用いることの第一のメリットとして計算量が厳密な計算よりも大幅に削減することができることが挙げられる。また計算結果が実験に適合するようにパラメータをきめるので場合によっては原理計算よりも実験に近い計算結果を得ることができる。本研究では炭素の混成軌道で表される結合を transferrable に表現できる Tersoff potential と分子間結合をよく表現するのに有効である Lennard Jones 型のポテンシャルを用いた。このポテンシャルを用いて カーボンナノチューブの構造最適化を行った。ここでは用いた経験ポテンシャルの関数形とパラメータを説明する。

2.1 Tersoff potential

Tersoff ポテンシャルは、炭素や珪素そして水素の共有結合を表現できる原子ポテンシャルである。ポテンシャル関数はモース型ポテンシャルを基に、多体の効果を斥力項と引力項にそれぞれ係数を付加した構成で、原子間の距離と 原子の周りにある原子の角度を変数とした関数になっている。この関数のパラメータは液体の炭素原子に関して第一原理計算の局所密度汎関数法の計算の結果にフィッティングされている [16]。

2.1.1 Tersoff potential の関数形

系の全結合エネルギー V は原子 i と原子 j の結合エネルギー Φ_{ij} の和で表される。

$$V = \frac{1}{2}\sum_{i \neq j} \Phi_{ij}, \tag{2.1.1}$$

この Tersoff ポテンシャルでは一般に $\Phi_{ij} \neq \Phi_{ji}$ であり、原子 i と原子 j 間の結合エネルギーは $(\Phi_{ij} + \Phi_{ji})/2$ と表現されることに注意する必要がある。

$$\Phi_{ij} = f_c(r_{ij})\{f_R(r_{ij}) - b_{ij} f_A(r_{ij})\}, \tag{2.1.2}$$

$f_R(r_{ij})$ と $f_A(r_{ij})$ の項はモース型とよばれる原子対ポテンシャルである。r_{ij} は (i,j) 原子間の距離であり、$f_R(r_{ij})$ は斥力項、$f_A(r_{ij})$ は 引力項を表している。

$$f_R(r_{ij}) = Aexp(-\lambda_1 r_{ij}), \tag{2.1.3}$$

$$f_A(r_{ij}) = Bexp(-\lambda_2 r_{ij}), \tag{2.1.4}$$

f_c は Tersoff potential を第一近接原子間ポテンシャルに制限する関数である。このため全エネルギー V の計算は、(i,j) の組に関して第一近接のみ計算に考慮すればよいので原理的に $O(N)$ の計算量である。$R+D$ の距離から、$R-D$ の距離までをサイン曲線で重みづけをしている。

$$f_c(r_{ij}) = \left\{ \begin{array}{ll} 1, & r_{ij} \leq R-D \\ \frac{1}{2} - \sin\frac{\frac{\pi}{2}(r_{ij}-R)}{D}, & R-D < r_{ij} < R+D \\ 0, & R+D \leq r_{ij} \end{array} \right\}, \tag{2.1.5}$$

又 b_{ij} は、多体の効果を表現する係数である。b_{ij} は引力項の係数として与えられる。

$$b_{ij} = (1+\zeta_{ij}^{\eta})^{-\delta}, \tag{2.1.6}$$

ζ_{ij} が多体の関数形を決めている。分子構造が sp^3 や sp^2 や sp の時に極小になることで、多体の効果を表現している。

$$\zeta_{ij} = \sum_{k \neq i,j} f_C(r_{ij}) g(\theta_{ijk}) \tag{2.1.7}$$

$g(\theta_{ijk})$ が結合角による効果を表現している。θ_{ijk} は、結合 ij と結合 ik とのなす角で k は、ij 原子以外の原子を表している。

$$g(\theta_{ijk}) = a(1 + c^2/d^2 - c^2/[d^2 + (h - \cos\theta_{ijk})^2]), \tag{2.1.8}$$

2.1.2 パラメータ

ここに C-C 結合の Tersoff ポテンシャルのパラメータを示す。Tersoff ポテンシャルの パラメータにはいくつかのバリエーションがある。代表的なのが Tersoff のパラメータ [16] と、Breener のパラメータ [17] である。このポテンシャルは他に水素やシリコンに関してのパラメータがあり、それぞれの混合体を表現するのに関数の平均をとるのだが、それぞれにいくつかのパラメータがあり、組合せる原子によって最適な組合せがある。ここでは 炭素のパラメータだけを示す。

表 2.1: Tersoff ポテンシャルパラメータ

	Tersoff[16]	Brenner[17]
$A[eV]$	1393.6	518.3696
$B[eV]$	346.7	328.0206
$\lambda_1[\mathring{A}^{-1}]$	3.4879	2.409357
$\lambda_2[\mathring{A}^{-1}]$	2.2119	1.867718
η	0.72751	1.0
δ	0.687276	0.80469
a	1.5724×10^{-7}	0.011304
c	38049.0	19.0
d	4.384	2.5
h	-0.57058	-1.0
$R[\mathring{A}]$	1.9	1.85
$D[\mathring{A}]$	1.0	1.5

2.1.3 パラメータの選択

岡田等は Tersoff ポテンシャル Tersoff パラメータによる C_{60} の構造が 2 つ結合長の比が NMR による結果と一致することを示した。そしてまた 結合長の修正パラメータに 0.963 を適用することにより、 C_{60} の構造を再現できることをしめした。本論文ではこれにならい Tersoff のパラメータを使用し、カーボンナノチューブ構造最適化を行った。

2.2 Lennard Jones potential

希ガス原子や球とみなせる分子の間の力は分子間の距離 r の関数であり、 r が小さいと強い斥力、 r が大きいと弱い引力 (いわゆる van der Waals 力)、強い斥力は電子同士の軌道が重なる効果表した物である。 Lennard Jones potential では一般に

$$U(R) = -\frac{C_m}{r^m} + \frac{C_n}{r^n} \tag{2.2.9}$$

とポテンシャルを近似的に表される。よく使われるのが $(6, 12)$ ポテンシャルである。

$$U(R) = 4\epsilon \left\{ -(\frac{\sigma}{r})^6 + (\frac{\sigma}{r})^{12} \right\} \tag{2.2.10}$$

もともと気体などのモデルに用いられるが、球状でない大きな分子の場合でも、原子対ポテンシャルとしてもちいてよい結果が得られる。

2.2.1 Lu の パラメータ

Lu らは C_{60} の結晶を表現するのに Lennard Jones 型のポテンシャルを用いたが、そのパラメータはグラファイトの層構造を表現することのできるよう決められた。六

角格子の面 (結合長 1.421Å) とし、 AB 積層のときに面間隔 3.354Å で極小値をとり、弾性定数 C_{33} (2次微分) をグラファイトの値を満たすようにパラメータをフィッティングした。[3]。その計算に用いたグラファイトのパラメータは Blakslee の論文によっている。[18] その Lennard Jones potential の パラメータは $\sigma = 3.407[\text{Å}]$ 、 $\epsilon = 2.968[\text{meV}]$ で、グラフファイト結晶 (AB staking) におけるの面間隔 $3.354[\mathring{A}]$、そして弾性率 $c_{33} = 4.08[\text{GPa}]$ を表現している。本論文ではこれ以下、 Tersoff と Lu のポテンシャルパラメータを用いる。

第 3 章

極小点探索

この章では、本研究で用いている極小点探索法について述べる。本研究では、カーボンナノチューブ分子などのの安定構造を求める必要がある。安定構造は、系のエネルギーを原子の位置の関数として、エネルギーが極小点になっている時の構造である。この時全ての変数に関しての 1 次偏微分が 0 である。極大値の場合も偏微分は 0 であるが、計算の初期値は、極小の位置の方が近い所を用いるので実際の計算おいては結果が極大に向かうことはない。

ある n 次の関数の極値をとる変数の値を求めたい時、解析的に解を求めるのは困難である。そこで、全原子の位置をあるベクトルの方向に動かして最小点を探し、また方向を変えて最小点を探す、 n 次元ランダムウォークを繰り返す事により極小点をことが考えられる。しかし、闇雲に探索しては収束が遅いし、収束が保証されている訳でない。。共役勾配法はその探索の繰り返し方を最適化する手法である。また分子動力学的手法は原子に運動を与えてやり、少しずつ運動量を減少させることにより、極小点を探す手法である。これらの手法を説明し利点 · 欠点について説明する。そして本研究で 周期的境界条件を課していないカーボンナノチューブの構造最適化を効率良く行う方法として分子動力学的手法を用いた構造最適化アルゴリズムを紹介する。そしてこれらの構造最適化のアルゴリズムでは ポテンシャル関数の原子の位置に関しての微分を行う必要がある。そこで本研究で用いた数値微分のアルゴリズムの導出をおこなったので紹介する。

3.1 共役勾配法 (Conjugate Graduate Method)

共役勾配法とは、 n 元 2 次関数の極点を $O(N)$ で求める手法である。構造最適化に応用するには、一般の 2 次関数でないポテンシャルを 2 次関数で近似して、その場合の極小点を求めることを繰り返すことによってエネルギーの最小の点を求めることができる。また 1 回共役勾配法を適用する場合の計算量が $O(N)$ であるので計算量が削減できる。

3.1.1 ２次関数近似

共役勾配法が適用出来る関数形は n 元２次関数であり、 $\mathbf{A}$ を $n \times n$ 正方行列、 $\mathbf{x}$ を n 次のベクトル、 c を定数とする。行列 $\mathbf{A}$ は 係数行列、ベクトル $\mathbf{b}$ は 係数ベクトルである。これは２次関数近似の係数は関数を数値的に微分する事により得ることができる。 3.4章で２次関数近似の方法を説明している。

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \cdot \mathbf{x} + c \tag{3.1.1}$$

この n 元２次関数の極小値を与える $\mathbf{x}$ は 次の式を満たす。

$$f' = \mathbf{A} \cdot \mathbf{x} - \mathbf{b} = 0 \tag{3.1.2}$$

この条件を満たす $\mathbf{x}$ を求めるアルゴリズムとして共役勾配法というアルゴリズムがある。

3.1.2 アルゴリズム

初期ベクトルを $\mathbf{x}_0$ とする。 $\mathbf{r}$ は 残差ベクトルといい、 $\mathbf{p}$ を修正ベクトルと呼ぶ。このなかで a は１次元最小値探索にもちいる最小位置を示す係数で、 c は一時変数である。

と $\mathbf{p}$

$$\mathbf{p_0} = \mathbf{r_0} = \mathbf{x_0} \tag{3.1.3}$$

$$a_i = \frac{||\mathbf{r}_i||^2}{\mathbf{p}_i \cdot \mathbf{A} \cdot \mathbf{p}_i} \qquad (i = 0, 1, \cdots) \tag{3.1.4}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + a_i \cdot \mathbf{p}_i \quad , \quad \mathbf{r}_{i+1} = \mathbf{r}_i - a_i \cdot \mathbf{A} \cdot \mathbf{p}_i \tag{3.1.5}$$

$$c_{i+1} = \frac{||\mathbf{r}_{i+1}||^2}{||\mathbf{r}_i||} \quad , \quad \mathbf{p}_{i+1} = \mathbf{r}_{i+1} + c_i \mathbf{p}_i \tag{3.1.6}$$

この手順で $\mathbf{x}_{i+1}$ を作ってゆくと、 すくなくとも n 回の反復で解を得ることができる。実際はもっと少ない反復回数で収束する。つまりここ部分に費される時間は実際の計算においては $\mathrm{O}(0)$ であり、実際は数値微分を行う所で費されている。

3.2 分子動力学的手法 (Molecular dynamical Mathod)

分子動力学的手法による構造最適化とは、原子間にかかる力を原子間ポテンシャルを数値微分することにより原子間にかかる力を計算し、ある瞬間の位置・速度の変化をもとめて、速度を一定の減衰率で減らすことにより、系の安定構造を求める手法である。

3.2.1 系の運動方程式

系の原子数を N 、原子の質量を m、各原子の座標を $\mathbf{r}_n$、各原子の速度を $\mathbf{v}_n$、各原子にかかる力を $\mathbf{F}_n$、系の全エネルギーを U とする。

$$\mathbf{r}_n = (x_n, y_n, z_n), \quad (1 \leq n \leq N), \tag{3.2.7}$$

原子の座標は位置ベクトルであらわされる。

$$U = U(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \cdots, \mathbf{r}_N), \tag{3.2.8}$$

系のエネルギー U は、位置ベクトル $\mathbf{r}_n$ の関数である。

$$\mathbf{F}_n = (\frac{\partial U}{\partial x_n}, \frac{\partial U}{\partial y_n}, \frac{\partial U}{\partial z_n}), \tag{3.2.9}$$

各原子がポテンシャルにより受ける力は系のエネルギー U を原子の位置において微分する事により求められる。

$$\frac{d\mathbf{v}_n(t)}{dt} = \frac{\mathbf{F}_n}{m}, \tag{3.2.10}$$

$$\frac{d\mathbf{r}_n(t)}{dt} = \mathbf{v}_n, \tag{3.2.11}$$

そして、運動方程式により 位置と速度と力が関係づけられている。

3.2.2 動力学シミュレーション

先の方程式を微小な時間 Δt 間隔で区切って考える。 N_{step} をシミュレーションのステップ数、Tを系における時間とする。

$$T = \Delta t N_{step} \tag{3.2.12}$$

$$\mathbf{v}_n(T) = \mathbf{v}_n(T - \Delta t) + \frac{\mathbf{F}_n(T)}{m}\Delta t \tag{3.2.13}$$

$$\mathbf{r}(T) = \mathbf{r}(T - \Delta t) + \mathbf{v}_n(T)\Delta t \tag{3.2.14}$$

Δt が十分に小さい場合、この式で原子の運動を近似することができる。

3.2.3 構造最適化に適用

式 (3.2.13) にパラメータ ε $(0 \leq \varepsilon \leq 1)$ を加えることにより、系の原子の運動速度を次第に減衰させることができる。

$$\mathbf{v}_n(T) = \varepsilon \mathbf{v}_n(T - \Delta t) + \frac{\mathbf{F}_n(T)}{m} \Delta t \tag{3.2.15}$$

$\varepsilon = 1$ の場合、系のエネルギー（ポテンシャルエネルギー＋運動エネルギー）は保存されるが、$(0 \leq \varepsilon < 1)$ の場合、ポテンシャルエネルギーは運動エネルギーへ移り極小に向かい、運動エネルギーは 0 に向かう。このこと利用して構造最適化が行える。実際の計算に用いる Δt と ε に関して述べる。Δt は大きいと原子の移動量が大きくなり最適化が早まるが、原子の移動量が大きくなり過ぎると、原子の位置が発散する。本研究で用いた 炭素の原子ポテンシャルの場合は 3.0[fs] を用いた場合にしばしば原子の位置が発散した。$\Delta t = 2.5[fs]$ で収束速度に関して良好な計算結果を得た。ε に関しては本論文では 0.9 をもちいているがもっと小さな値でもよい。

3.3 C_{60} と ナノチューブへの応用

共役勾配法 (CG 法) と分子動力学的手法 (MD 法) を用いて C_{60} とナノチューブを構造最適化を行なったときの比較を行なう。

3.3.1 C_{60} の構造最適化

C_{60} の最適化を CG 法と MD 法で行なった。ポテンシャルは 岡田らが示した C_{60} の結合長を再現するよう修正した Tersoff ポテンシャルを用いた。初期条件は、結合距離は全て 1.42Å である。今回の MD 法での time step は 2.5[fs] 、減衰パラメータは 0.9 、初期速度 は 0 で行っている。

図 3.1 は、ある WS での実際の最適化の過程であり、縦軸が炭素 1 個当たりの結合エネルギー、横軸が計算時間 である。 Tersoff ポテンシャルと CG 法による C_{60} の構造計算は岡田等が行っており、収束解は 2 つの手法に差異はなく共通の結果を得た。single ボンドは 1.46Å、 double bond は 1.39Å である。共役勾配法の場合は滑らかにエネルギーが減衰しているのに対して MD 法は振動しつつ減衰しているのが分る。MD 法は 共役勾配法に必要な 2 次微分がない分有利であるが、著しくメリットが有るとは言えない。 MD 法は 系に適合するパラメータを必要とする分汎用性が低い。

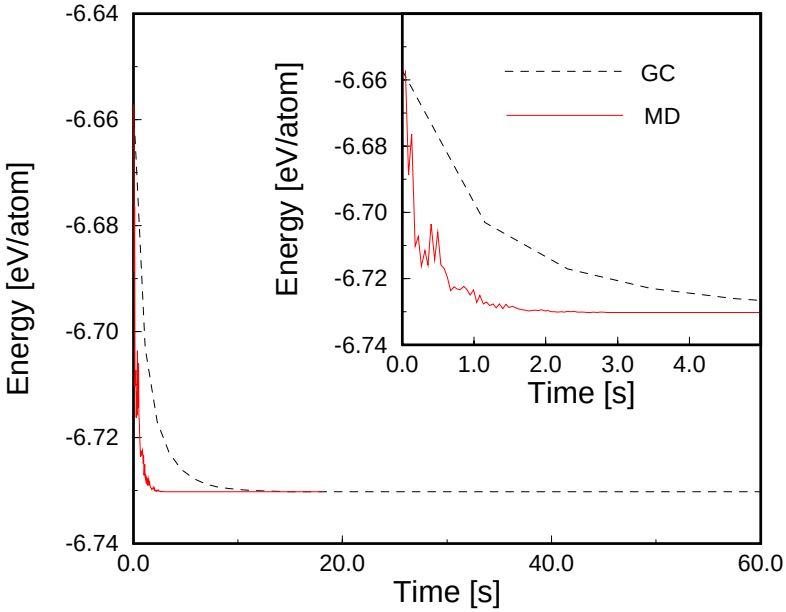

図 3.1　C_{60} の最適化の系のエネルギーと計算時間 [1]

3.3.2　ナノチューブ

(10,10) の チューブを 25 ユニットセル積んだものの最適化を実行した。原子数は 1000 である。

今回の MD 法での time step は 2.5[fs] 、減衰パラメータは 0.9 を用いている。

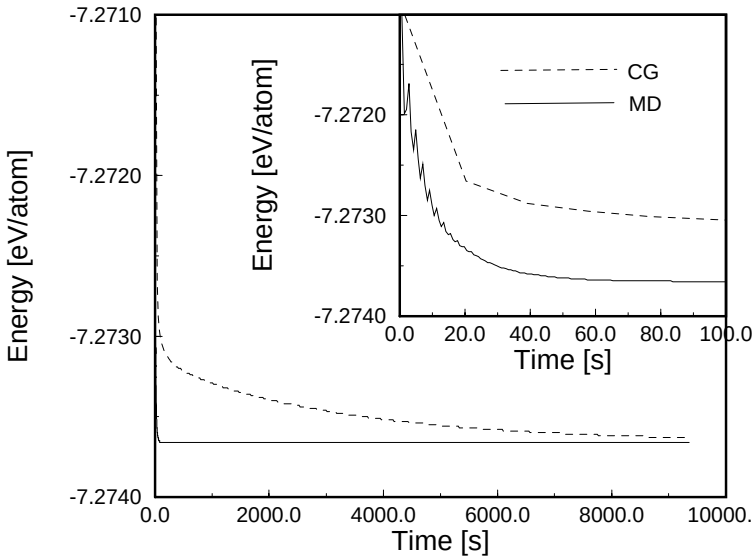

図 3.2　最適化の過程 [2]

それぞれの計算 step における収束の仕方が C_{60} の場合と異なり、 MD 法では振動しつつ最適化されるのは変わらないが、共役勾配法の収束の速度が著しく鈍化してい

[1] 図 3.1 = m99ryou/c60time-2.xvgr
[2] 図 3.2 = m99ryou/25x10-10time-2.xvgr

る。また実際の計算時間は、 共役勾配法が 2次微分を必要とするのと収束の速度の鈍化により、実用的な時間での収束解を得る事が出来ない。

共役勾配法での収束の鈍化の原因であるが、チューブの構造が 軸方向に長く 対称性が高いことが挙げられる。チューブの軸方向では ポテンシャルエネルギーから受ける力がつりあってしまう。よって、最初のステップで端の原子のみが最適化されるが中心部分の原子は動かないことになる。動く原子が各 step で少数になるため収束が鈍化する。

つまり通常カーボンナノチューブのような1次元的構造の共役勾配法による構造最適化は周期的境界条件を課して行われ、チューブの長さが変わらない条件で、そのユニット内での原子の位置のみの最適化を行うので困難は生じない。。しかし今回のようなチューブ切片を構造最適化においてはチューブの長さが変化するような系では、ユニットセル内だけでの原子再配置だけではチューブの長さを変えることができない。つまり軸方向の最適化に困難が生ずる。

MD 法では振動しつつ原子の最適化が行なわれるので、安定位置にない原子は力がつりあっていても動くことができる。それゆえ動いている原子が 共役勾配法と比べると格段に多い。例えて言うならチューブをゆすりながら弛緩させている状態である。

3.4 数値微分法・2次関数近似

この節では、本研究で用いている数値微分法について述べる。微分とは、式 (3.4.16) で表されるように極限の形式で表されるが、この定義の近似として、極限を取る操作の代わりに有限の微小値をもちいる方法がある。この方法を数値微分という。また、関数をある点で2次関数に近似する事が出来る。ここでは微分値とその係数の関係を述べる。

3.4.1 数値 1 次微分

x の 関数 $f(x)$ の 微分 $f'(x)$ の定義は、無限小のシンボルとして ε をもちいて、

$$f'(x) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x)}{\varepsilon}, \tag{3.4.16}$$

と表される。ここで、ある x における $f(x)$ の 微分値を求めたいとき、解析的な導関数より求めることが出来るが、導関数を導出しないで数値的に求める数値微分の定義式を示す。微分の定義は $\varepsilon \to 0$ という極限の操作を行なうが、この操作を “十分小さな有限の値” に置き換えることで、数値的に関数の微分値を計算できる。式 (3.4.16) は、関数が連続であるなら、

$$f'(x) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x-\varepsilon)}{2\varepsilon}, \tag{3.4.17}$$

であっても同等である。ここで、極限の操作を取り払って、

$$f'(x) = \frac{f(x+\varepsilon) - f(x-\varepsilon)}{2\varepsilon} \qquad : \varepsilon\text{は微小値}, \tag{3.4.18}$$

が、数値 1 次微分の定義式である。式 (3.4.17) の形にするのは、極限の時と違い有限な ε を持つことによる x 軸での ε の誤差を生じるのを防ぐためである。

3.4.2 数値 2 次微分

x の関数 $f(x)$ の 2 階微分 $f''(x)$ の定義は、無限小のシンボルとして ε ε' をもちいて、

$$f''(x) = \lim_{\varepsilon' \to 0} \frac{f'(x+\varepsilon') - f'(x)}{\varepsilon'}, \tag{3.4.19}$$

と定義出来る。式 (3.4.16) より、

$$f''(x) = \lim_{\varepsilon \to 0} \lim_{\varepsilon' \to 0} \frac{f(x+\varepsilon+\varepsilon') - f(x+\varepsilon) - f(x+\varepsilon') + f'(x)}{\varepsilon\varepsilon'}, \tag{3.4.20}$$

また 2 変数関数 $f(x,y)$ の 2 次微分の場合、

$$\frac{\partial^2}{\partial x \partial y} f(x,y) = \lim_{\varepsilon \to 0} \lim_{\varepsilon' \to 0} \frac{f(x+\varepsilon, y+\varepsilon') - f(x+\varepsilon, y) - f(x, y+\varepsilon') + f(x,y)}{\varepsilon\varepsilon'}, \tag{3.4.21}$$

と定義できる。これを 数値的な定義に直す。式 (3.4.20) と式 (3.4.21) は それぞれ

$$f''(x) = \lim_{\varepsilon \to 0} \lim_{\varepsilon' \to 0} \frac{f(x+\varepsilon+\varepsilon') - f(x+\varepsilon-\varepsilon') - f(x-\varepsilon+\varepsilon') + f'(x-\varepsilon-\varepsilon')}{4\varepsilon\varepsilon'}, \tag{3.4.22}$$

$$\frac{\partial^2}{\partial x \partial y} f(x,y) = \lim_{\varepsilon \to 0} \lim_{\varepsilon' \to 0} \frac{f(x+\varepsilon, y+\varepsilon') - f(x+\varepsilon, y-\varepsilon') - f(x-\varepsilon, y+\varepsilon') + f(x-\varepsilon, y-\varepsilon')}{4\varepsilon\varepsilon'}, \tag{3.4.23}$$

と定義しても同等である。これを数値的定義に置き換えると、

$$f''(x) = \frac{f(x+\varepsilon+\varepsilon') - f(x+\varepsilon-\varepsilon') - f(x-\varepsilon'+\varepsilon') + f'(x-\varepsilon-\varepsilon')}{4\varepsilon\varepsilon'}, \tag{3.4.24}$$

$$\frac{\partial^2}{\partial x \partial y} f(x,y) = \frac{f(x+\varepsilon, y+\varepsilon') - f(x+\varepsilon, y-\varepsilon') - f(x-\varepsilon, y+\varepsilon') + f(x-\varepsilon, y-\varepsilon')}{4\varepsilon\varepsilon'}, \tag{3.4.25}$$

ここで、微小量 $\varepsilon, \varepsilon'$ に適当な値をもちいる事により微分値を得る事が出来る。しかし注意すべき点がある、一般的には $\varepsilon \neq \varepsilon'$ である場合は成り立つが、$\varepsilon = \varepsilon'$ の場合もしくはそれ近い状態では、式 (3.4.24) において $f(x+\varepsilon-\varepsilon') - f(x-\varepsilon+\varepsilon')$ が 0 になり、このことは 次で示す 2 次関数近似の定義と矛盾する。

3.4.3 ２次関数近似

関数をある点の付近にて多項式関数に近似する事が出来る。本研究では共役勾配法においてポテンシャル関数を２次関数近似をする。ある関数を２次関数に近似したときの係数は次のように求められる。式 (3.4.26) のような関数がある時、各係数を求める式はつぎの通りである。

$$f(x,y) = ax^2 + by^2 + cxy + dx + ey + g, \tag{3.4.26}$$

ここで、適当な数値 ε を考えて、

$$a = \frac{f(x+2\varepsilon,y) - f(x+\varepsilon) - f(x-\varepsilon,y) + f(x-2\varepsilon,y)}{6\varepsilon^2}, \tag{3.4.27}$$

$$b = \frac{f(x,y+2\varepsilon) - f(x,y+\varepsilon) - f(x,y-\varepsilon) + f(x,y-2\varepsilon,y)}{6\varepsilon^2}, \tag{3.4.28}$$

$$c = \frac{f(x+\varepsilon,y+\varepsilon) - f(x+\varepsilon,y-\varepsilon) - f(x-\varepsilon,y+\varepsilon) + f(x-\varepsilon,y-\varepsilon)}{4\varepsilon^2}, \tag{3.4.29}$$

$$d = \frac{f(x+\varepsilon,y) - f(x-\varepsilon,y)}{2\varepsilon}, \tag{3.4.30}$$

$$e = \frac{f(x,y+\varepsilon) - (x,y-\varepsilon)}{2\varepsilon}, \tag{3.4.31}$$

$$g = f(0,0), \tag{3.4.32}$$

これら関係は、式 (3.4.26) を実際に代入して解けば間単に示せる。これらの式を、数値微分の式と比べると、

$$f''(x) = 3a/2, \quad \frac{\partial^2}{\partial x \partial y} f(x,y) = c, \quad f'(x) = d \tag{3.4.33}$$

ここで、式 (3.4.27) は、式 (3.4.24) において、 $\varepsilon \to \frac{3}{2}\varepsilon$, $\varepsilon' \to \frac{1}{2}\varepsilon$ の場合である。この時、ここの２次関数近似における係数の求め方は数値微分の定義と一致する。

$$f''(x) = 2a \tag{3.4.34}$$

本論文では、２次関数近似の係数をこの 3.4.3節で示した方法で求めた。

3.5 近接原子探索の最適化

本研究で用いられるポテンシャルを扱う時、あらかじめ、ある原子の隣の位置にある原子の表を作成することは計算量の減量、プログラムの可視性を向上するのに有効である。しかし、原子ネットワークの全ての原子同士のペアを対象に近接判定をすることは、計算量が膨大になるので望ましくない。この章では、全ての原子ペアを対象にせずに原子対 (i,j) のリストを作成する方法を説明する。

3.5.1 近接判定

ある原子対の 距離を r とする。この距離が ある値以下であるならば近接であると判断する。炭素原子の場合は、 sp^2 の場合は、第一近接の距離は、 1.42Å であり、第二近接の距離は 2.4Å である。近接の判定はその中間の値をもちいる。 Tersoff Potential のカットオフ距離 2.1Å はそのようにして決められた。実際に計算される系では、ある原子に対して 近接であるのは高々数個である。ほとんどの原子はカットオフ距離以上離れているので、ここで Tersoff potential や Lennard jones potential の計算を省くことができる。しかし全ての原子対の距離を計算して判定を行うと計算量が $O(N^2)$ となる。

3.5.2 メッシュ化

リスト作製の計算量を $O(N)$ にする方法として、計算される空間をあらかじめ分割しておき、原子を分割された空間にそれぞれ割り当てることにより、遠い原子同士 の判定を省くことを考える。あるブロックの内部と隣接だけを考慮すればよい分割方法は、空間を 1 辺がある大きさ以上立方体のブロックに分割することである。このときあるブロックの隣接ブロックは 26 個あるが、中心のブロックからみて、隣接ブロックを越えた所のブロックに近接原子が無い条件は、ブロックの一辺が近接判定距離よりも大きいことである。本論文において近接判定の距離は Tersoff potential と Lennard Jones potential のそれぞれのカットオフ関数の値が 0 になる距離を用いた。

3.5.3 微分における計算の省き方

Tersoff ポテンシャルの関数を微分を計算する時は、どの範囲の原子間ポテンシャルを考慮すべきなのか述べる。

1 次微分

Tersoff ポテンシャルは、原子同士の結合エネルギー表現している。ある原子を原子 A と呼ぶ。原子 A からは結合の手が何本かでているが、その結合の手のエネルギーに関連する原子は、原子 A の近接原子と原子 A の第二近接原子である。。原子 A の近接原子群を B 原子群、 B 原子群の近接原子群 C(原子 A と 原子群 B 自身を除く) とする。実際の計算に応用するには、まず第一近接原子群 B のリストベクトル LIST1

を作る。又第二近接の原子群 C のリストベクトル LIST2 は LIST1 から作ることができる。

2 次微分

２次微分の時は少し複雑になるが次の 3 通りの場合に分けられる。微分に関する変数が同一原子の位置変数である場合と、隣り合う原子に別れている場合合と第二近接に別れている場合がある。同一原子内にある場合は、 1 次微分と同じ原子を考慮すればよい。隣り会う原子同士の場合は、 2 つの原子の共通の近接原子だけを考慮する。第二近接同士の場合は、２つの原子の共通の第二近接原子だけを考慮する。そうすることにより全ての原子対のポテンシャルを計算せずに 2 次関数近似を行う事ができる。

3.6 最適化構造

この章では、これまで述べた、共役勾配法、分子動力学的手法 及び、数値微分法を用いて、カーボンネットワークの最適化構造について述べる。

3.6.1 経験的原子ポテンシャルによる最適化 C_{60} 構造

Tersoff ポテンシャルと 共役勾配法による C_{60} の 構造最適化は 岡田等によってなされたが、ここでは、プログラムの検証を兼ねて 同じ計算を行う。 C_{60} には 2 種類の結合が 現われる。 single bond と double bond であり それぞれ、 NMR による測定では 1.447 $\AA$ と 1.398 $\AA$ である。素の Tersoff potential の計算では single-bond は 1.502Å、 1.460 Å になる。実験による値よりは大きい値ではあるがそれらの 比が正しくなる、岡田等は それらを 調整する為に係数として 0.963 を長さに適用することにより、 C_{60} の構造を 再現した [10]。我々が作成したプログラムも 同様の結果を得た。

3.6.2 経験的原子ポテンシャルによる最適化カーボンナノチューブ構造

岡田等が C_{60} 等のフラーレンを 構造最適化 したのを応用してカーボンナノチューブの構造最適化を行なう。しかしここで、共役勾配法によるカーボンナノチューブの構造最適化 が C_{60} と同様に行なうことが出来ないことがわかった。そこで、あらたな計算手法をして分子動力学的手法 (MD 法) を 導入した。詳しくは3章 極小点探索法を参照。

分子動力学的手法では、最適化方向へ移動してゆく原子の分布が早い段階で中心付近の原子にも及ぶので、軸方向の最適化が共役勾配法よりも有利になる。

3.7 経験的原子ポテンシャルによる最適化グラファイト構造

グラファイトの構造を Tersoff potential と、 Lennard Jones potential をもちいて表現する事を考える。

3.7.1 グラフェンとダイヤモンド構造の再現

Tersoff ポテンシャルでのグラフェンの最近近接原子間距離は、1.467Å になる。またダイヤモンド構造の最近近接原子間距離は 1.548Å である。しかし実際の最近接距離は、それぞれ グラフェンが 1.421Å、ダイヤモンド構造が、1.544Å である。これは Tersoff ポテンシャルの LDA 計算への原子間距離のフィッティグが sp_3 を基準になされていることが挙げられる。

Tersoff ポテンシャルが sp sp_2 sp_3 をどれでも再現するようフィッティングされているが、それぞれでの原子間距離にずれを生じている。本研究において カーボンナノチューブの結合の種類はほとんどが sp_2 であるので、 sp_2 の結合に注目してパラメータを調整する。参考に C_{60} の場合では 0.963 を与えると、 NMR よる C_{60} の 結合距離を再現する。 CNT においては構造がグラファイトに近いと考えられるので、 sp_2 の結合距離 1.421Å を再現するよう 0.9685 を用いる。チューブの直径が小さい (3.39Å) 場合においては 0.963 大きなチューブの極限では 0.9685 になる。今回はチューブに対しても 0.9685 をパラメータに適用している。

層構造の再現

このポテンシャルでは、結晶構造 (AB stack) を 再現することが出来る。 (AB Stack) での面間 3.354$\mathring{A}$ で系のエネルギーが極小を持つ。また不安定な構造 (AA stack) を固定して面間隔を変化させた時の エネルギー極小の面間は 3.44 $\mathring{A}$ をとる。つまり、ポテンシャルは AA と AB stack 構造の中間に Turbostratic 構造を持つことが言える。また グラファイトの弾性定数に関しては Blakslee の論文に [18] に詳しい。

第 4 章

ナノチューブへのグラファイトパッチ吸着のカイラリティ依存性

ナノチューブへの グラファイト小片の吸着を考える事は、カーボンナノチューブの生成過程に関わる事であり、興味深い問題である。カーボンナノチューブの成長モデルは、最初の種チューブにグラファイトの小片がつぎ当てするようにまとわりついて新たな層が成長するパッチモデルと、培地になる金属からチューブが生えて根本から成長するというモデルがある。本章では、グラファイトのパッチがチューブにどのように吸着するかを計算する。とくにチューブのカイラル角と吸着角度の関係について考えたい。今回は、チューブの半径は同じ位でカイラル角の異なる チューブに C_{24} や C_{54} を吸着させて、その安定配置や吸着エネルギーを計算した。

4.1 計算方法

4.1.1 計算対象

計算対象として、 (10,10) チューブの半径に近い半径 6.8Å のチューブ と 炭素クラスターに C_{24} と C_{54} の 2 つを用いる。

4.1.2 計算手順

半径 6.8Å 付近のチューブとして、 (10,10) (12,8) (13,7) (14,5) (16,2) (17,0) の物を用意する。チューブの長さは 50Å 以上となるよう、数ユニットセル重ねている。グラファイトクラスタ C_{24} と C_{54} のジオメトリを用意する。そしてそれぞれ単体で構造最適化を行なう。チューブのそばにグラファイトクラスタを 3.4Å 離して配置するが、それぞれクラスタを回転させて角度で 1 ° づつ回転させたデータを用意して、構造最適化を行い、吸着エネルギーと置いた時の角度のグラフを作成する。アームチェア型のチューブにたいしてグラファイトでいう AB stacking になる配置を基準に角度 0 ° と定めている。吸着エネルギーは構造最適化後の全エネルギーからクラスターとチューブが無限遠に離れているときの全ネルギーを引いた値である。構造最適化の方法は MD 法を用い、構造最適化のパラメータは timestep 2.5fs 減衰率 0.9 を用いる。構造最適化の step 数を 200 とした。

4.2 炭素クラスタ吸着エネルギとチューブへの吸着角

チューブのカイラル角が吸着角度へ与える影響をしらべる。

4.2.1 C_{24} 吸着エネルギとチューブへの吸着角

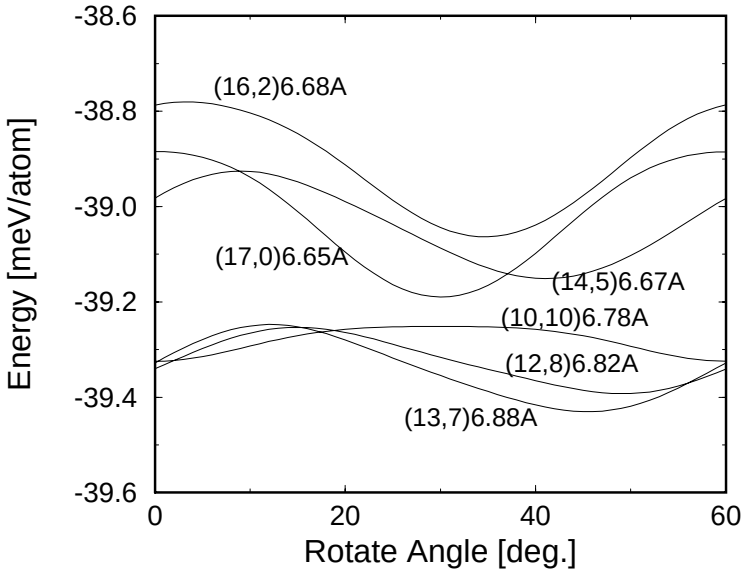

図 4.1 C_24 吸着のエネルギーと吸着角度 [1]

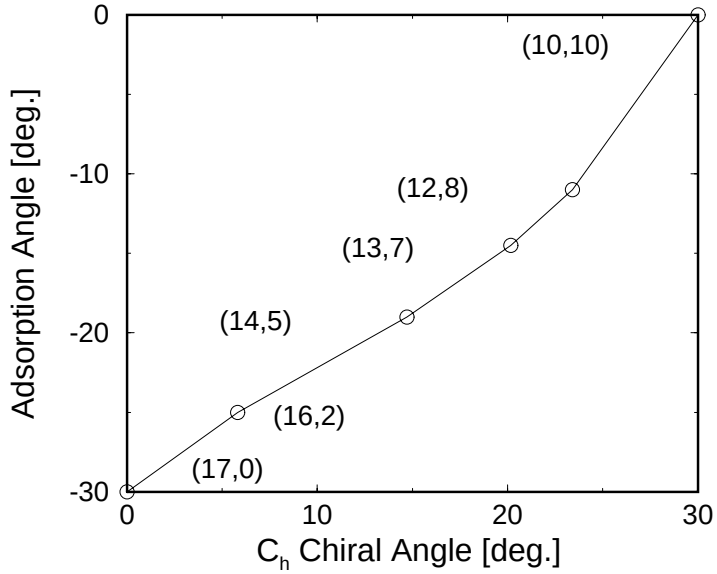

図 4.2 チューブのカイラル角とエネルギー極小な角度 [2]

エネルギー極小である安定な角度は、グラファイトでいう AB stacking になっていいることが、チューブのカイラル角の変化の仕方と対応していることから分る図 4.2。

[1] 図 4.1 = m99ryou/r68c24-3.xvgr
[2] 図 4.2 = m99ryou/angle-angle-c24-3.xvgr

図 4.1をみると、カイラル角の変化に応じて吸着角度が同じく変化する様子は、今回計算したチューブでは一様であるが、グラフのエネルギーの平均の高さがそれぞれ異なっている。これは、チューブの直径が大きいほど吸着エネルギーが大きくなることに対応している。理由は分らないが、それぞれのチューブの直径に依存して、振幅が大きくなっている。

4.2.2 C_{54} 吸着エネルギとチューブへの吸着角

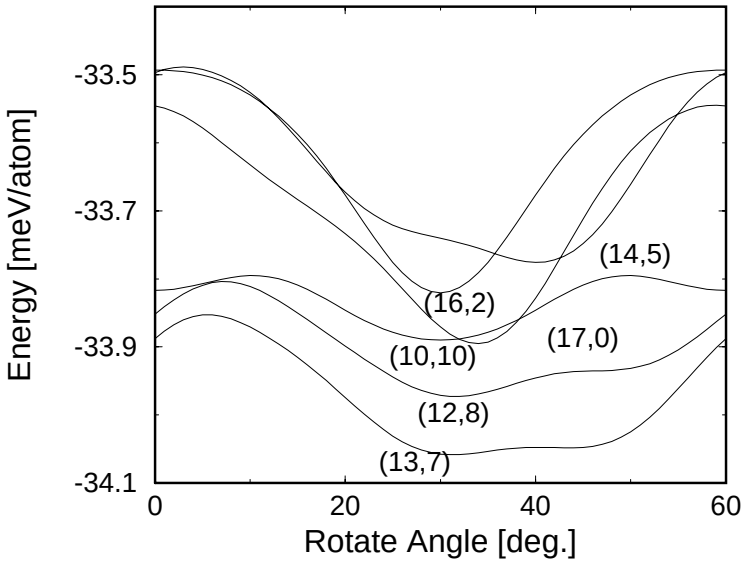

図 4.3 半径 6.8Å [3]

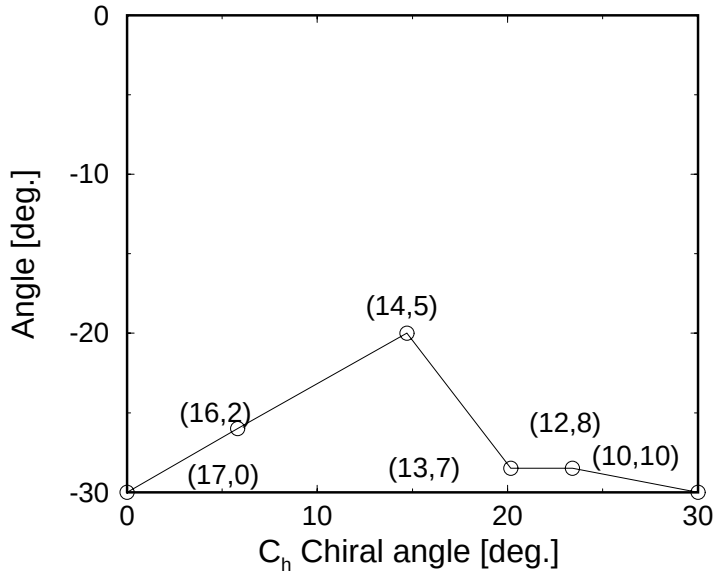

図 4.4 カイラル角と安定角 [4]

[3]図 4.3 = m99ryou/r68c54-3.xvgr
[4]図 4.4 = m99ryou/angle-angle-c54-3.xvgr

安定な角度が C_{24} の時と異なり、必ずしもカイラル角に依存していない。どのチューブでも 30 度の所に極小になる項が存在している。

C_{54} の場合にカイラル角に依存しない 30 度安定の項があることを考える。クラスタが大きい程チューブの直径が小さい程チューブのカイラル角に関わらず、吸着クラスタがチューブに対してジグザク型に吸着するのが有利になっている。クラスタの大きさの効果による影響の可能性があるがチューブの直径との関連もあると思われる。チューブの直径変化に関する依存性を計算する必要がある。

4.3 チューブの直径変化と吸着エネルギーと吸着角度の関係

カーボンクラスターがチューブに吸着するエネルギーの角度依存性において 30 度安定項が存在する。この 30 度安定項はクラスタの大きさに依存するのかチューブの直径に依存するのかを調べる。

4.3.1 C_{24} C_{54} と (n,n) チューブ

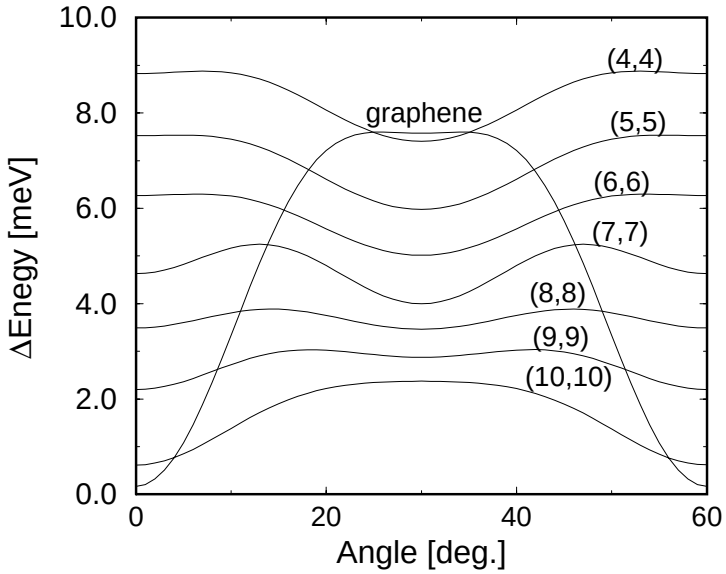

図 4.5 C_{24} 吸着エネルギーとチューブの直径 [5]

[5]図 4.5 = m99ryou/nn-c24-3.xvgr

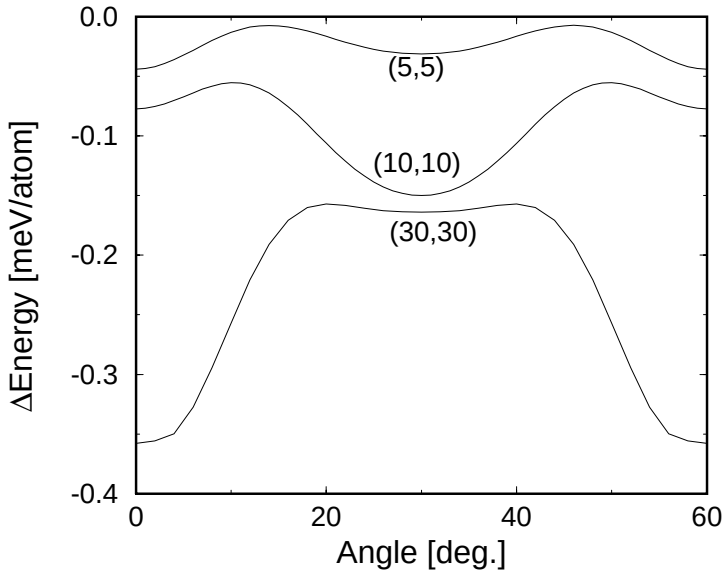

図 4.6 C_{54} 吸着エネルギーとチューブの直径 [6]

C_{24} の場合、チューブの直径が大きい時は 30 度安定項は見られないが直径が小さくなるにつれて 30 度安定項が現われる。ポテンシャルの平らな部分がチューブの直径が小さいん場合３０°から 離れた所に現れているが、チューブの直径が増えるにしたがって減少している。C_{54} の場合、チューブの直径に関わらず 30 度安定項が見られるが、チューブの直径が大きい時は小さく、またチューブが細すぎても 30 度安定項の振幅は小さくなる。チューブの直径がおおきくなると角度 30 ° のポテンシャルの形が平になることが示されている。

4.3.2 考察

C_{24} にもエネルギーの角度依存性に 30 度安定項が現れることから、エネルギーの角度依存性の 30 度安定項はチューブの直径によることがいえる。このことからチューブの直径と角度 30 の関わりは、チューブの丸まった構造つまり曲率のある構造に対して、積み重なり方は AB stacking とは関わらず、クラスタの向きがジクザグの方向がチューブの軸方向に向かうときに接触面積が大きくなることがいえる。チューブの直径が大きくなると、曲率が小さくなるのでグラファイトとクラスターの関係近づくので、安定角は クラスタの向きが チューブとの関係が AB stacking になるよう決まる。

4.4 チューブの直径変化と吸着エネルギーの関係

チューブの吸着エネルギーは チューブの直径の依存性が強くカイラリティの依存性は小さいので、チューブはアームチェア型だけを考慮する。

[6]図 4.6 = m99ryou/nn-c54-3.xvgr

4.4.1 チューブの直径変化と吸着エネルギー

アームチェア型チューブとグラファイトクラスターの安定配置における吸着エネルギーをチューブカイラルベクトル n の 1/2 乗でプロットした。

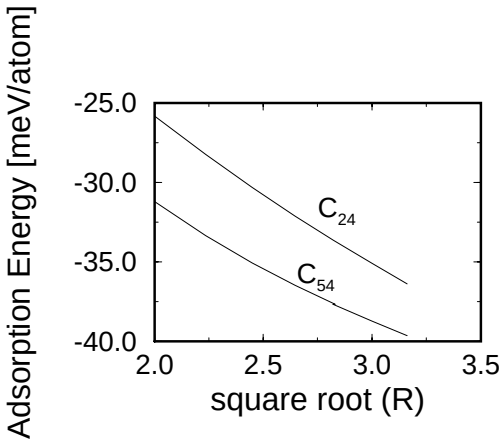

図 4.7 吸着エネルギー変化とチューブの直径 [7]

4.4.2 考察

まず注目すべき点は、吸着エネルギーが 半径の 1/2 乗に比例する点である。この関係は Lu らの グラファイトとナノチューブの相互作用エネルギー関係と基本的に同じである。Lu らはグラファイトの表面にナノチューブを吸着させたときの吸着エネルギーはチューブの直径の 1/2 乗に比例することを示してた。今回の計算は グラファイトの表面でなくグラファイトの小片であるが、吸着エネルギーはチューブの直径の関係は同じ傾向を示している。

C_{54} の吸着エネルギーが C_{24} のそれよりも小さいのは、チューブとの原子数当たりの接触面積が小さくなるからである。チューブの直径が大きくなるにつれて、半径の 1/2 乗 の関係からずれるがこれは、最終的には グラフイトとクラスターの関係に近づくことになるので飽和することに起因する。この飽和の仕方は real-log プロットをすると 直線になることから、実際の関数形は log(x) である。グラファイト表面とグラファイトパッチの違いはパッチの大きさが有限であることが要因になっていると考えられる (図 4.8)。

[7] 図 4.7 = m99ryou/nn-e-3.xvgr

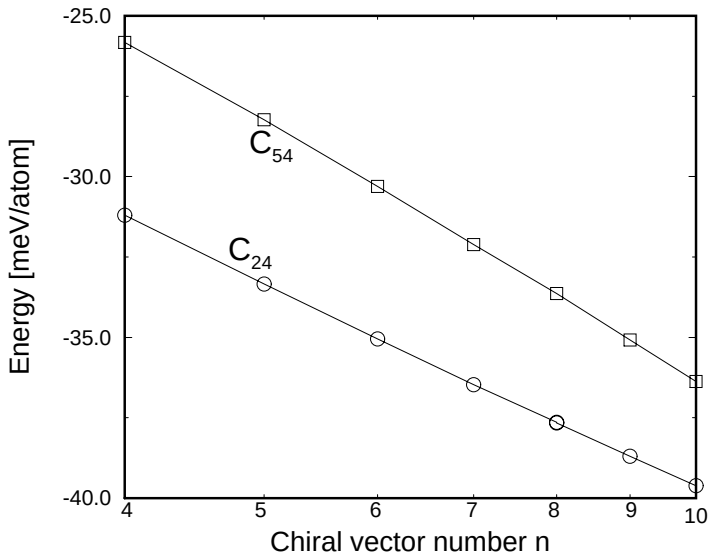

図 4.8 吸着エネルギー変化とチューブの直径 (横軸を log プロット) [8]

4.5 まとめ

この章ではチューブにグラファイトの小片の吸着する様子を解析した。吸着角度に関わる要素をチューブの直径・カイラル角、そしてカーボンクラスタパッチの大きさとしたが、それらの要素はどれも吸着角度に大きく影響を与えることが分った。とくにチューブの直径の変化で０°安定と３０°安定という２つの安定構造を示すことは予想外のことであった。またポテンシャルの構造でポテンシャルの形が平になる

この計算においては、カーボンクラスターをプロブーにしてポテンシャル構造を探っているが、この計算を 次章の２層ナノチューブの面間ポテンシャル構造の理解を助けることになる。

[8]図 4.8 = m99ryou/nn-energy-2.xvgr

第 5 章

２層構造ナノチューブの安定構造のカイラリティ依存性

２層構造チューブの層間ポテンシャル構造を Lennard Jones ポテンシャル (2.2章) を用いて計算する。計算上の問題点として、一般のナノチューブの場合カイラリティの異なるチューブの軸方向の格子定数は整合しない (incommensurate) ので周期的な境界条件での計算ができない。そのため層間ポテンシャル構造を計算するために、２層チューブの内側のチューブの長さを十分長くとり、外側は 1 ユニットセルだけの構造に関して計算を行った (図 5.1)。内側のチューブの長さは、層間相互作用の端の効果による乱れが無視できるようにするため、端同士の距離は相互作用のカットオフ距離よりも離れている必要がある。断熱ポテンシャル構造を、内側チューブの軸方向 (格子定数)・軸回転運動 (360 °) に対してのポテンシャルの値の変化として表した。

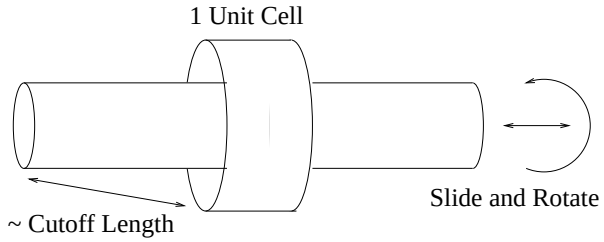

図 5.1 計算に用いる 2 層チューブの構造 [1]

5.1 計算に用いた 2 層チューブの立体構造

2 層チューブの内側チューブのカイラリティとして 60x(5,5), 8x(6,4), 8x(6,-4), 35x(9,0), 25x(8,2), 25x(8,-2) を用いた。カイラルベクトルの前に示している nx はユニットセルを n 個分並べてあるかを示している。 それぞれのチューブは数ユニットセル並べて 140Å 以上の長さの立体構造にしている。チューブ同士の端の効果を無視できるよう端同士の距離が層間ポテンシャルのカットオフ長 (17.5Å) よりも長くする。内側の

[1]図 5.1 = m99ryou/2ltube.eps

チューブを格子定数分動かしても端同士の距離が 20 Å 以上長くなるよう長さを 140Å 以上にした。これらを Tersoff ポテンシャルのみで構造最適化を行った。

また、外側のチューブとして内側と外側の半径の差が 3.4Å 程度になるチューブを選ぶ。A.1 から (17,0) 6.65Å から (16,4) 7.17Å のチューブを選んだ。それらの構造最適化されたユニットセルの立体構造を計算する。但し、ユニットセルの格子定数が極端に短いアームチェア型とジグザグ型のチューブは、長さを 24Å 以上にするため、それぞれ 10 ユニットセル、8 ユニットセルの長さの立体構造を用意した。

これらのチューブを組み合わせて、スライドと回転の変化と Lennard Jones ポテンシャルの値 (断熱ポテンシャル) を計算する。ポテンシャルエネルギーは、層間相互作用の全ポテンシャルを外側チューブの原子１個当たりの値に規格化した。

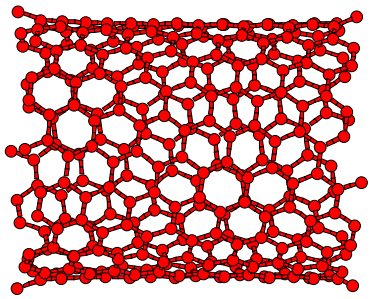

図 5.2　1x(14,5) チューブの端の広がり [2]

ところで、チューブの構造最適化を行うにあたり、チューブの切口の構造が開いてしまう (図 5.2)。この開いた構造がポテンシャルの形状に影響を与える可能性がある。そのためこの端の効果が計算の障害にならないよう、切口に緩衝構造を追加して構造最適化を行い (図 5.3)、内部の構造を取りだして端が広がっていない最適化構造のユニットセルを作成した。

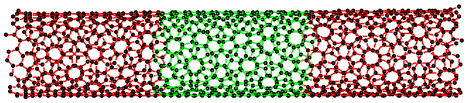

図 5.3　端に緩衝チューブを付けた 3x(14,5) チューブの最適化構造 [3]

こうして、2 層構造チューブの断熱ポテンシャルを計算した結果の１例を示す。図 5.4 は (6,4) と (10,10) チューブの組合せの場合の断熱ポテンシャルで、横が回転運動、縦が軸方向の移動を示し、明るい所はポテンシャルエネルギーが高く、逆に暗い所はエネルギーが低いことを示している。横の回転は図の端から端が１回転を意味し、縦の端から端は内側のチューブの軸方向の格子定数分の層がずれることを意味し、この場合 (6,4) チューブの格子定数の 1.86[nm] である。また下部にある数値はポテンシャルエネルギーの最大値と最小値を示している。

[2]図 5.2 = m99ryou/tube-edge.eps
[3]図 5.3 = m99ryou/3x14-5o.eps

図 5.4　(6,4)-(10,10) チューブの断熱ポテンシャル形状 [4]

図 5.4では ポテンシャルが、ボルトとナットの関係になっていることが考えられる。その障壁の大きさはおおよそ 30[meV] であることが分る。今回計算した結果の画像は A.2 にまとめて掲載しておく。また、計算した結果のデータは ./m99ryouma/data/ に置いてある。

5.2　外側のチューブのユニットセル数依存

本章では 外側のチューブのユニットセルは原則として 1 ユニットセルをもちいているが、2 ユニットセルにした場合に計算結果が大きく異ならないことをここで示す。ここでは 内側を (6,4) チューブとして、外側を (14,5) と (16,4) チューブの場合を示す。それぞれ格子定数が (14,5) は 2.42[nm] 、 (16,4) は 0.65[nm] である。

図 5.5　(6,4)-(14,5) チューブのポテンシャル形状
(左: 1 ユニットセル, 右: 2 ユニットセル) [5]

[4]図 5.4 = m99ryou/2lbmp/06041010.eps
[5]図 5.5=m97ryou/2lbmp/06041405.eps,06041405-2.eps

図 5.6　(6,4)-(16,4) チューブのポテンシャル形状
(左: 1 ユニットセル, 右: 2 ユニットセル) [6]

まず、一見して分るのは、ポテンシャルの形状が (14,5) と (16,4) のどちらでもユニット数を変えることによって著しい変化は見られないことである。つまりこのポテンシャル形状はそれぞれ 2 つのチューブの組合せの固有のものであることが言えて、外側チューブの長さの効果は小さい事が分る。

5.3　面内相互作用エネルギーと面間距離

2 層チューブのそれぞれの組合せにおいて、軸方向と軸回転の運動におけるポテンシャルエネルギーが最小になるところが安定構造である。ここでは、その安定構造におけるポテンシャルエネルギーの値を、チューブの半径の差としてプロットした (図 5.7)。

5.3.1　結果

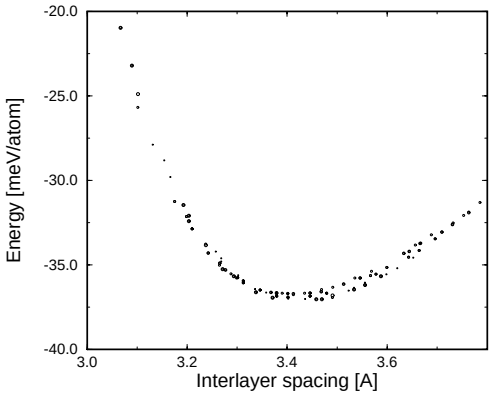

図 5.7　安定構造のポテンシャルエネルギーと層間隔 [7]

まず今回計算したチューブの組合せにおいて、層間隔が 3.4Å の付近でエネルギーが最小になる傾向が見られる。

[6] 図 5.5 = m99ryou/2lbmp/06041604.eps,06041604-2.eps
[7] 図 5.7 = m99ryou/all-r-2.xvgr

どのカイラリティのペアにおいても 面間距離が 3.4Å の時が最安定構造のエネルギーが極小になる。チューブのカイラリティによる依存性は見られない。また、チューブの直径は大きい程エネルギー的に安定な傾向が見られる。

5.3.2　考察

どのカイラリティのペアにおいても面間距離が 3.4Å の時が最安定構造のエネルギーが極小になるということは、エネルギー的に安定である 2 つのチューブの組合せはは面間の距離が大きな要因で、その大きさが 3.4Å になるチューブの組合せであることを示している。また全てのグラフのポテンシャルの形は 面間相互作用をモデル化した Jennard Jones ポテンシャルの形をそのまま表しておりチューブのカイラリティの効果は小さいことがいえる。これらの結果は、 MWNT に良く見られる面間隔が 3.4Å より少し大きい所にあることとは矛盾せず、 MWNT の生成過程において面間相作用の効果が小さくないことと関係がある。またチューブの吸着エネルギーに対して、軸方向や軸回転したときのポテンシャルの変化は小さい (吸着エネルギー 30m[eV] に対して 1m[eV] よりも小さい変化) のでカイラリティの影響はないと考えられる。

5.4　回転や軸方向の摺動に対するエネルギー障壁

エネルギーの極小になるチューブの組合せは面間の距離が 3.4Å であった。しかしそれぞれの組においての断熱ポテンシャルの振幅は面間の距離に関しては無関係であるようだ。そこで、ここでは、それぞれのチューブの組合せでの断熱ポテンシャルの最小値と最大値の差の大きさは何によって決まるかを考えてみたい。図 5.8のグラフは、それぞれ内側のチューブを固定して外側のチューブを変えたときの振幅の大きさと面間の距離でプロットしたものである。特に振幅の大きいものには チューブのカイラリティを示している。

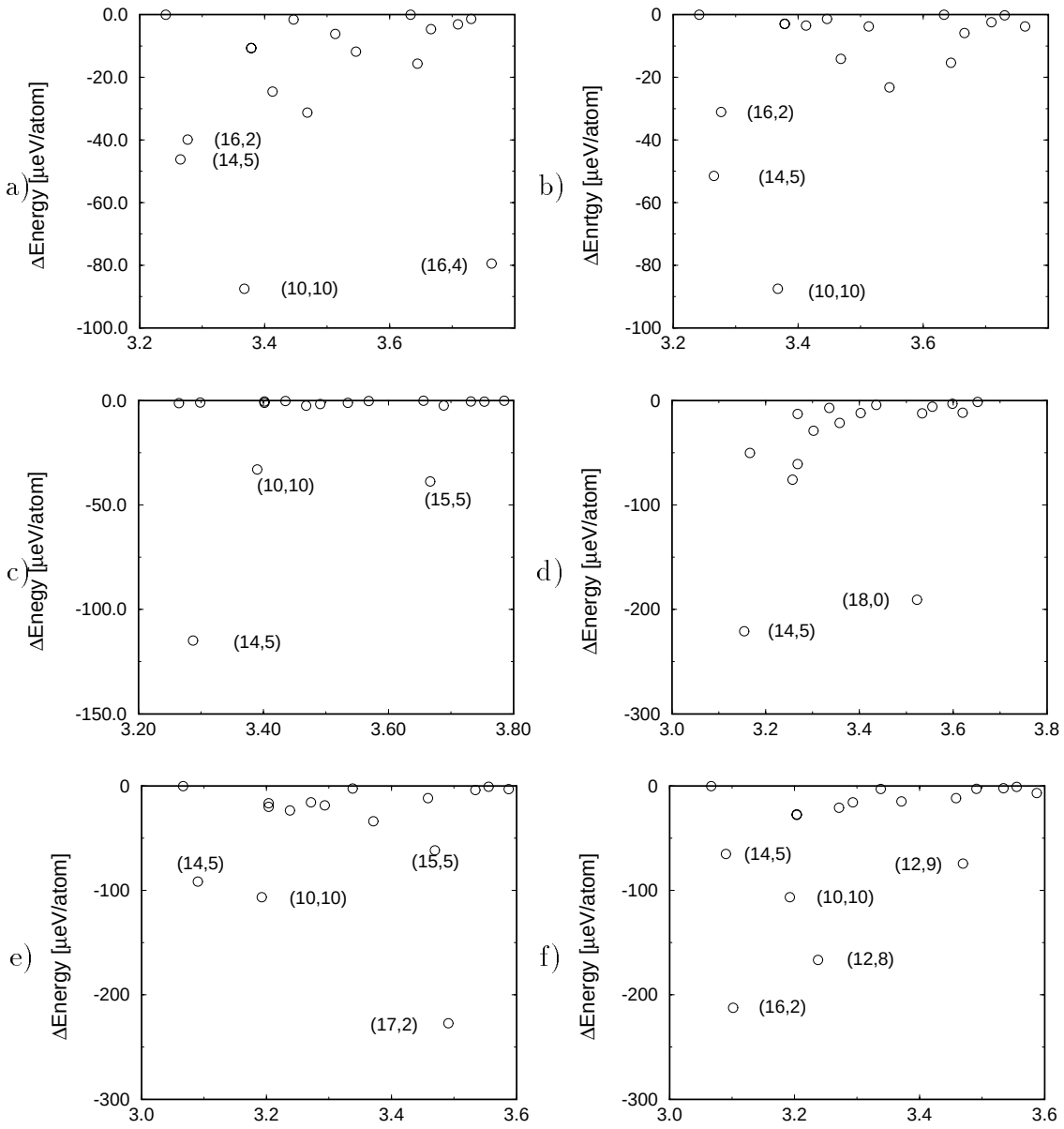

図 5.8　ポテンシャルエネルギーの高低差

a-(6,4) b-(6,-4) c-(5,5) d-(9,0) e-(8,2) f-(8,−2)　[8]

まず一見して分ることには、エネルギー障壁の高さは面間距離には依存していないことが分る。このことは振幅は面間隔以外の要素が大きいことを示している。その要素とは何であるかを知りたい。特定のカラルベクトルのチューブのみに障壁があらわれている。内側がアームチェア型かジグザク型の場合はエネルギー障壁大きいものと小さなものがはっきりと分れているが、内側がカイラルチューブの場合は少しバラけている。

もう少し詳しく考えてみる。内側が $(5,5)$ チューブや $(9,0)$ チューブの場合に注目すると、エネルギーの障壁が大きい組合せと、小さな組合せがはっきり分れている。

[8]図 5.8=m99ryou/2lbmp/0604-r-3,0406-r-3.xvgr, 0505-r-3.xvgr, 0900-r-3.xvgr, 0802-r-3.xvgr, 0208-r-3.xvgr

この計算をする前にはこのような小さな振幅は予想できなかった。大きな振幅 (数十 meV、これもエネルギーの値としては小さいが) の大きさは前章のグラファイトパッチの振幅と同じ程度であるので理解できる。

面間相互作用の振幅が小さくなる理由として、チューブの軸方向の格子定数が、不整合であるので 2 つのチューブの長さが長い程平均化されて振幅が小さくなることは考えられるが、不整合なチューブのチューブの組合せでも振幅はあまり変化していないのでそれが理由ではない。またチューブの円周方向関しては周期的であるので、振幅に関しては強まる場合と弱まる場合があると考えられる。

ここで、前章 4.3.1のグラファイトパッチにに置ける振幅の形状に注目したい。図 4.5 のグラフでは (4,4)(5,5)(6,6) チューブのとき ３０° から離れた部分において、グラフの形状が平になっている。。振幅の小さくなる要素としてこのグラファイトパッチの場合の平な部分が関係していると思われる。つまり、振幅の小さくなるチューブの組合せは、このグラファイトパッチのグラフの平な部分の関係と同じになっていて、大きな振幅の部分は３０° 付近の傾きが大きい所の関係になっていると思われる。

5.4.1　考察

特定のチューブペアにおいて、チューブのスライドや回転のエネルギー障壁が大きくなることは、興味深い結果である。チューブのボルトとナットの関係あるチューブのペアが存在することが示された。この計算においてエネルギー障壁はエネルギーが最大と最小の位置の差で表しているが、実際エネルギーポテンシャルとしては 最小点か鞍点を通る所を調べる必要がある。

5.5　まとめ

この章の計算で得られたことはとしてまず、エネルギー的に安定なチューブの組合せは面間隔が 3.4Å になるチューブの組で、チューブカイラリティによるポテンシャルエネルギーの変化は小さいので、チューブカイラル角は影響を与えていないことが分った。しかし吸着エネルギーよりも小さいとはいえ、その振幅はチューブのカイラリティに大きく依存することがしめされた。

とくに振幅の小さな組合せがあるが、特異な振幅の消失はグラファルイトパッチが吸着するときのポテンシャル形状と関連があることを示唆することができた。

エネルギー障壁の大きさ
エネルギー障壁の高さは面間距離には依存せず、また特定のカイラリティの組の時に大きくなる。

第 6 章

結論

本研究ではカーボンナノチューブの層間の構造について経験的ポテンシャルをもちいて解析を行った。その過程で、 Tersoff ポテンシャルを用いたカーボンナノチューブの構造最適化について共役勾配法と分子動力学的手法による構造最適化アルゴリズムに比較を行うことによって、 “大きな有限の長さのカーボンナノチューブの構造最適化の手法は共役勾配法より、分子動力学的手法を用いるほうが、原子数が多い場合計算量において有利” であることが分った。

また、カーボンナノチューブにグラファイトの小辺が吸着するときの安定な構造を計算した。そこではチューブの直径やカイラル角また、小辺の大きさを変えて計算したが、グラファイト小片の安定な配置というのは単純には決まらないことが分った。これらに要素は吸着構造に大きな影響を与えているそこで現れるポテンシャル形状、とくに平らになる部分や傾きが急である部分が存在するが、２層チューブのチューブの相対位置の変化に対する断熱ポテンシャルの変位の大小を決める要素は、それらの部分の足し合わせにあると考えられる。それゆえ、特定のカイラリティのペアの螺旋構造に応じてエネルギーの振幅が大きくなったり、小さくなったりすることを説明できると考えられる。

チューブの軸方向の不整合によるポテンシャル形状の平均化は大きな影響を与えず、２つチューブの螺旋構造のに応じたエネルギーのポテンシャルの振幅が存在することが大きな成果であると考える。

本論文ではカーボンナノチューブの構造最適化で、層内の結合と層間の結合を分離しているが、この分離の仮定が正しいのか検証する必要がある。今回は２層の構造を用いているが さらに多層になるとチューブが変形することも考えられるし、またチューブの面間が 3.4Å よりも大きく小さい場合の結合長の変化もどのような影響を与えるか分っていない。また、単層ナノチューブの構造最適化において、結合長がカイラリベクトルに依存して変化する。この変化を詳しく調べることも重要であると考えられる。

謝辞

本研究及び論文作成に当たり、御指導を賜わりました指導教官の齋藤理一郎助教授に厚く御礼の言葉を申しあげます。また研究室セミナー等にてさまざまな御指導を賜わりました、木村忠正教授、湯郷成美助教授、一色秀夫助手に感謝致します。そして、パート秘書をされています山本 純子さんに感謝致します。また、２年間研究活動をともにしてきました沼 知典さんに感謝致します。そして、勉強や遊びに一緒にすごしてきた、木村・湯郷研究室の学生の皆様に感謝の意を表したいと思います。また、既に卒業された竹谷 隆夫さん、八木 将志さんら先輩方にお世話になりましたことを感謝します。そして、経済的援助と生活を支えくださった私の両親に感謝申し上げます。

参考文献

[1] Yahachi Saito and Tadanobu Yoshikawa, Interlayer spacings in carbon nanotubes, Phys. Rev. B **48**, 1907 (1993).

[2] Shunji Bandow, Radial Thermal Expansion of purified Multiwall Carbon Nanotubes Measured by X-ray Diffraction, Jpn. J. Appl. Phys. **36**, 1404–1405 (1997).

[3] Jian Ping Lu and *et al*, Ground State and Phase Transitions in solid C_{60}, Phys. Rev. Lett. **68**, 1551 (1992).

[4] Jian Ping Lu and Weitao Yang, The shape of large single- and multople-shell fullerenes, Phys. Rev. B **49**, 11421–11424 (1994).

[5] Jakyoung Song and R. R. Cappelletti, Lennard-Jones-potential calculation of C_60 stage-one graphite, Phys. Rev. B **50**, 14678–14681 (1994).

[6] Jian Ping Lu, Elastic Properties of Carboon Nanotubes and Nanoropes, Phys. Rev. Lett. **79**, 1297–1300 (1997).

[7] A. Buldum and Jian Ping Lu, Atomic Scale Sliding and Rolling of Carbon Nanotubes, Phys. Rev. Lett. **83**, 5050–5053 (1997).

[8] J.-C.Charlier and J.-P. Michenaud, Energetics of Multilayered Carbon Tubeles, Phys. Rev. Lett. **70**, 1858–1861 (1997).

[9] Adam H. R. Palser, Interlayer interactions in graphite and carbon nanotubes, Phys. Chem. Chem. Phys. **1**, 4559–4464 (1999).

[10] Susumu Okada, Electronic and Geometric Structure of Fullernenes and Polymerized C_{60}, (Nov 1997). 東京工業大学 博士論文.

[11] S. Iijima, Nature **354**, 56 (1991).

[12] H .Hamada, S. Sawada, and A. Oshiyama,

[13] J.M. Mintmire, B.I. Dunlap, and C. T. White, Phys. Rev. Lett. **68**, 631 (1992).

[14] R.Saito, M. fujita, G. Dresselhaus, and M. S. Dresselhaus, Phys. Rev. B **46**, 1804 (1992).

[15] 竹谷 隆男. カーボンナノチューブの構造, Feb 1996. 卒業論文 電子工学科 UEC.

[16] J. Tersoff, Empirical Interatomic Potentuial for Carbon, with Applications to Amorphous Carbon, Phys. Rev. Lett. **61**, 2879 (1988).

[17] Brenner, Phys. Rev. B **42**, 9458 (1990).

[18] O. L. Blakslee, D. G. Proctor, E.J. Seldin, G. B. Spence, and T. Weng, Elastic Constance of Compression-anialed Pyrolytic Graphite, J. Appl. Phys. **41**, 3373–3382 (1977).

付録 A

計算データ

A.1 直径順に並べた カーボンナノチューブのカイラリティ

炭素の結合距離を 1.42Å と仮定した場合のとカイラリティ $\mathbf{C}_h$ と直径 $\mathbf{d}_h$ を列挙する。直径の単位は [nm] 半径の単位は [deg.] とする。表は直径順に並べてある。計算は ./diameter-chiral.f を用いた。

$\mathbf{C}_h$	d_h[nm]	θ[deg.]	$\mathbf{C}_h$	d_h[nm]	θ[deg.]	$\mathbf{C}_h$	d_h[nm]	θ[deg.]
(1, 0)	0.078	0.00	(8, 0)	0.626	0.00	(11, 1)	0.903	4.31
(1, 1)	0.136	30.00	(7, 2)	0.641	12.22	(10, 3)	0.923	12.73
(2, 0)	0.157	0.00	(8, 1)	0.669	5.82	(12, 0)	0.939	0.00
(2, 1)	0.207	19.11	(5, 5)	0.678	30.00	(7, 7)	0.949	30.00
(3, 0)	0.235	0.00	(6, 4)	0.683	23.41	(11, 2)	0.949	8.21
(2, 2)	0.271	30.00	(7, 3)	0.696	17.00	(8, 6)	0.952	25.29
(3, 1)	0.282	13.90	(9, 0)	0.705	0.00	(9, 5)	0.962	20.63
(4, 0)	0.313	0.00	(8, 2)	0.718	10.89	(10, 4)	0.978	16.10
(3, 2)	0.341	23.41	(6, 5)	0.747	27.00	(12, 1)	0.981	3.96
(4, 1)	0.359	10.89	(9, 1)	0.747	5.21	(11, 3)	1.000	11.74
(5, 0)	0.391	0.00	(7, 4)	0.755	21.05	(8, 7)	1.018	27.80
(3, 3)	0.407	30.00	(8, 3)	0.771	15.30	(13, 0)	1.018	0.00
(4, 2)	0.414	19.11	(10, 0)	0.783	0.00	(9, 6)	1.024	23.41
(5, 1)	0.436	8.95	(9, 2)	0.795	9.83	(12, 2)	1.027	7.59
(6, 0)	0.470	0.00	(6, 6)	0.814	30.00	(10, 5)	1.036	19.11
(4, 3)	0.476	25.29	(7, 5)	0.817	24.50	(11, 4)	1.053	14.92
(5, 2)	0.489	16.10	(10, 1)	0.825	4.72	(13, 1)	1.059	3.67
(6, 1)	0.513	7.59	(8, 4)	0.829	19.11	(12, 3)	1.076	10.89
(4, 4)	0.542	30.00	(9, 3)	0.847	13.90	(8, 8)	1.085	30.00
(5, 3)	0.548	21.79	(11, 0)	0.861	0.00	(9, 7)	1.088	25.87
(7, 0)	0.548	0.00	(10, 2)	0.872	8.95	(10, 6)	1.096	21.79
(6, 2)	0.565	13.90	(7, 6)	0.882	27.46	(14, 0)	1.096	0.00
(7, 1)	0.591	6.59	(8, 5)	0.889	22.41	(13, 2)	1.104	7.05
(5, 4)	0.611	26.33	(9, 4)	0.903	17.48	(11, 5)	1.110	17.78
(6, 3)	0.621	19.11	(11, 1)	0.903	4.31	(12, 4)	1.129	13.90

$\mathbf{C}_h$	d_h[nm]	θ[deg.]	$\mathbf{C}_h$	d_h[nm]	θ[deg.]	$\mathbf{C}_h$	d_h[nm]	θ[deg.]
(14, 1)	1.137	3.42	(12, 9)	1.429	25.29	(18, 6)	1.694	13.90
(9, 8)	1.153	28.05	(16, 4)	1.435	10.89	(13, 12)	1.695	28.68
(13, 3)	1.153	10.16	(13, 8)	1.437	22.17	(20, 3)	1.695	6.89
(10, 7)	1.159	24.18	(14, 7)	1.450	19.11	(14, 11)	1.699	26.04
(11, 6)	1.169	20.36	(18, 1)	1.450	2.68	(15, 10)	1.706	23.41
(15, 0)	1.174	0.00	(17, 3)	1.463	7.99	(16, 9)	1.717	20.82
(14, 2)	1.182	6.59	(15, 6)	1.467	16.10	(19, 5)	1.717	11.39
(12, 5)	1.185	16.63	(16, 5)	1.487	13.17	(17, 8)	1.731	18.26
(13, 4)	1.205	13.00	(19, 0)	1.487	0.00	(20, 4)	1.744	8.95
(15, 1)	1.215	3.20	(11, 11)	1.492	30.00	(18, 7)	1.749	15.75
(9, 9)	1.220	30.00	(12, 10)	1.494	27.00	(13, 13)	1.763	30.00
(10, 8)	1.223	26.33	(18, 2)	1.494	5.21	(14, 12)	1.765	27.46
(11, 7)	1.230	22.69	(13, 9)	1.500	24.01	(15, 11)	1.770	24.92
(14, 3)	1.230	9.52	(14, 8)	1.510	21.05	(19, 6)	1.770	13.29
(12, 6)	1.243	19.11	(17, 4)	1.512	10.33	(16, 10)	1.778	22.41
(16, 0)	1.253	0.00	(15, 7)	1.524	18.14	(17, 9)	1.790	19.93
(13, 5)	1.260	15.61	(19, 1)	1.528	2.54	(20, 5)	1.794	10.89
(15, 2)	1.260	6.18	(18, 3)	1.540	7.59	(18, 8)	1.806	17.48
(14, 4)	1.282	12.22	(16, 6)	1.542	15.30	(19, 7)	1.824	15.08
(10, 9)	1.289	28.26	(12, 11)	1.560	28.56	(14, 13)	1.831	28.78
(11, 8)	1.294	24.79	(13, 10)	1.564	25.69	(15, 12)	1.834	26.33
(16, 1)	1.294	3.00	(17, 5)	1.564	12.52	(16, 11)	1.841	23.90
(12, 7)	1.303	21.36	(20, 0)	1.566	0.00	(20, 6)	1.846	12.73
(15, 3)	1.308	8.95	(14, 9)	1.572	22.85	(17, 10)	1.851	21.49
(13, 6)	1.317	17.99	(19, 2)	1.572	4.95	(18, 9)	1.864	19.11
(17, 0)	1.331	0.00	(15, 8)	1.583	20.03	(19, 8)	1.881	16.76
(14, 5)	1.336	14.70	(18, 4)	1.589	9.83	(14, 14)	1.898	30.00
(16, 2)	1.338	5.82	(16, 7)	1.599	17.27	(15, 13)	1.900	27.64
(10, 10)	1.356	30.00	(20, 1)	1.606	2.42	(20, 7)	1.900	14.46
(11, 9)	1.358	26.70	(17, 6)	1.618	14.56	(16, 12)	1.905	25.29
(15, 4)	1.358	11.52	(19, 3)	1.618	7.22	(17, 11)	1.913	22.95
(12, 8)	1.365	23.41	(12, 12)	1.627	30.00	(18, 10)	1.924	20.63
(17, 1)	1.372	2.83	(13, 11)	1.629	27.25	(19, 9)	1.938	18.35
(13, 7)	1.376	20.17	(14, 10)	1.635	24.50	(20, 8)	1.956	16.10
(16, 3)	1.385	8.44	(18, 5)	1.640	11.93	(15, 14)	1.967	28.86
(14, 6)	1.392	17.00	(15, 9)	1.644	21.79	(16, 13)	1.970	26.58
(18, 0)	1.409	0.00	(20, 2)	1.650	4.72	(17, 12)	1.976	24.32
(15, 5)	1.411	13.90	(16, 8)	1.657	19.11	(18, 11)	1.985	22.07
(17, 2)	1.416	5.50	(19, 4)	1.666	9.37	(19, 10)	1.998	19.84
(11, 10)	1.424	28.43	(17, 7)	1.674	16.47			

A.2 ２層カーボンナノチューブの断熱ポテンシャル濃淡プロット

図 1.1 (6,4)(17,0)*

図 1.2 (6,4)(14,5)

図 1.3 (6,4)(16,2)

図 1.4 (6,4)(10,10)

図 1.5 (6,4)(11,9)

図 1.6 (6,4)(15,4)

図 1.7 (6,4)(12,8)

図 1.8 (6,4)(17,1)

図 1.9 (6,4)(13,7)

図 1.10 (6,4)(16,3)

図 1.11 (6,4)(14,6)

図 1.12 (6,4)(18,0)*

図 1.13 (6,4)(15,5)

図 1.14 (6,4)(17,2)

図 1.15 (6,4)(11,10)*

図 1.16 (6,4)(12,9)

図 1.17 (6,4)(16,4)

図 1.18 (6,−4)(17,0)*

図 1.19 (6,−4)(14,5)

図 1.20 (6,−4)(16,2)

図 1.21 (6,−4)(10,10)

図 1.22 (6,−4)(11,9)

図 1.23 (6,−4)(15,4)

図 1.24 (6,−4)(12,8)

図 1.25 (6,−4)(17,1)

図 1.26 (6,−4)(13,7)

図 1.27 (6,−4)(16,3)

図 1.28 (6,−4)(14,6)

図 1.29 (6,−4)(18,0)

図 1.30 (6,−4)(15,5)

図 1.31 (6,−4)(17,2)

図 1.32 (6,−4)(11,10)

図 1.33 (6,−4)(12,9)

図 1.34 (6,−4)(16,4)

図 1.35 (5,5)(17,0)

図 1.36 (5,5)(14,5)

図 1.37 (5,5)(16,2)

図 1.38 (5,5)(10,10)

図 1.39 (5,5)(11,9)

図 1.40 (5,5)(15,4)

図 1.41 (5,5)(12,8)

図 1.42 (5,5)(17,1)

図 1.43 (5,5)(13,7)

図 1.44 (5,5)(16,3)

図 1.45 (5,5)(14,6)

図 1.46 (5,5)(18,0)*

図 1.47 (5,5)(15,5)

図 1.48 (5,5)(17,2)

図 1.49 (5,5)(11,10)

図 1.50 (5,5)(12,9)

図 1.51 (5,5)(16,4)

図 1.52 (9,0)(17,0)

図 1.53 (9,0)(14,5)

図 1.54 (9,0)(16,2)

図 1.55 (9,0)(10,10)

図 1.56 (9,0)(11,9)

図 1.57 (9,0)(15,4)

図 1.58 (9,0)(12,8)

図 1.59 (9,0)(17,1)

図 1.60 (9,0)(13,7)

図 1.61 (9,0)(16,3)

図 1.62 (9,0)(14,6)

図 1.63 (9,0)(18,0)

図 1.64 (9,0)(15,5)

図 1.65 (9,0)(17,2)

図 1.66 (9,0)(11,10)

図 1.67 (9,0)(12,9)

図 1.68 (9,0)(16,4)

図 1.69 (8,2)(17,0)

図 1.70 (8,2)(14,5)

図 1.71 (8,2)(16,2)

図 1.72 (8,2)(10,10)

図 1.73 (8,2)(11,9)

図 1.74 (8,2)(15,4)

図 1.75 (8,2)(12,8)

図 1.76 (8,2)(17,1)

図 1.77 (8,2)(13,7)

図 1.78 (8,2)(16,3)

図 1.79 (8,2)(14,6)

図 1.80 (8,2)(18,0)

図 1.81 (8,2)(15,5)

図 1.82 (8,2)(17,2)

図 1.83 (8,2)(11,10)

図 1.84 (8,2)(12,9)

図 1.85 (8,2)(16,4)

図 1.86 (8,−2)(17,0)

図 1.87 (8,−2)(14,5)

図 1.88 (8,−2)(16,2)

図 1.89 (8,−2)(10,10)

図 1.90 (8,−2)(11,9)

図 1.91 (8,−2)(15,4)

図 1.92 (8,−2)(12,8)

図 1.93 (8,−2)(17,1)

図 1.94 (8,−2)(13,7)

図 1.95 (8,−2)(16,3)

図 1.96 (8,−2)(14,6)

図 1.97 (8,−2)(18,0)

図 1.98 (8,−2)(15,5)

図 1.99 (8,−2)(17,2)

図 1.100 (8,−2)(11,10)

図 1.101 (8,−2)(12,9)

図 1.102 (8,−2)(16,4)

付録 B

付録 プログラムソース

ここでは、本論文で作成使用したプログラムを紹介する。

B.1 構造最適化プログラム

本研究において作成した、カーボン分子の構造最適化プログラムを示す。このプログラムは、炭素構造の３次元座標のファイルを読み込み、エネルギーの極小値を与える３次元座標を求め、標準出力に出力する。原子間ポテンシャルには、Tersoff potential Lennard Jones potential を用いている、また最適化手法に共役勾配法と 分子動力学的手法を用いる事ができる。

計算の繰り返し回数は

```
      K = 0
      DO 41 I = 1 , 2000
```

の 2000 を変える。

共役勾配法を用いる場合は ２微分を計算するところがコメントになっているので、コメント指示を外し、また１次構造最適化をコメントアウトする。

```
      DO 38 J = 1 , MAX_ATOM3
         DIFF2(0,J) = 0
         DD(J) = 0.0D0
 38   CONTINUE

       CALL CALC_DIFF2(N,ZAHYO,LIST,LIST2,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
       CALL CALC_DIFF2_VDW
     #      (N,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF2,DIFF2_DATA)


      CALL CONJGRAD(N,ZAHYO,DIFF1,DIFF2,DIFF2_DATA)
```

```
C       DO 39 J = 1 , N*3
C            VELO(J) = VELO(J) + DIFF1(J) * CONV_VELO
C           ZAHYO(J) = ZAHYO(J) + VELO(J)
C            VELO(J) = VELO(J) * 0.9D0
C
C 39    CONTINUE
```

計算を実行するには
初期座標データ tube.xyz を用意して、

```
md5 tube.xyz >! kekka.xyz
```

とする。

```
c
c     Tersoff potential for Carbon system
c     (1998/Jun/8)   By  R.Hatsuo .
c

      PROGRAH HD
      INCLUDE 'PARAHETER'

C     原子座標の FILE 名変数
      CHARACTER*50 FILENAHE
      CHARACTER*50 FILENAHE2

C     原子数の保持変数
      INTEGER N

C     loop 用変数
      INTEGER I,J,K

C     座標用配列
      REAL*8  ZAHYO(HAX_ATOH3)
      CHARACTER*8  AN(HAX_ATOH)

C     近接リスト配列
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)

C     第二 2 近接リスト配列
      INTEGER LIST2(0:HAX_LIST2_N,HAX_ATOH)

C     近接リスト配列
      INTEGER LIST_VDW(0:HAX_LIST_VN2,HAX_ATOH)

C     近接データ配列 IDX でリストから参照される
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      REAL*8  DIFF1(HAX_ATOH3)
      REAL*8  VELO(HAX_ATOH3)
      REAL*8  DD(HAX_ATOH3)


      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)

C     TERSOFF 関数の型
      REAL*8 TERSOFF

C     系のエネルギー
      REAL*8 ENERGY , ENERGY_VDW

      INTEGER IDX



C     FILE 名を取得する
      CALL GETFILENAHE(FILENAHE)

      open(unit=20,file='md.log',ACCESS='APPEND')
      write(20,25) FILENAHE

 25   format(a50)

C     XYZ 座標を読み込む
```

```
      CALL READ_ZAHYO(FILENAHE, N, ZAHYO,AN)

      DO 29 J = 1 , HAX_ATOH3
         VELO(J) = 0.0D0
 29   CONTINUE

      IDX = 0
      CALL HAKE_LIST(N,ZAHYO,LIST,KYORI,CUTOFF,IDX)
      CALL HAKE_LIST2(N,LIST,LIST2)

      CALL HAKE_LIST_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,IDX,LIST,
     #LIST2,AN)

      ENERGY = TERSOFF(N,ZAHYO,LIST,KYORI,CUTOFF)
      ENERGY_VDW = VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF)
      ENERGY = ENERGY + ENERGY_VDW

C      CALL PRINT_ZAHYO(N,ZAHYO,I,ENERGY,DIFF1,AN)

C      WRITE(*,*) ENERGY
C      STOP

      K = 0
      DO 41 I = 1 , 2000


C     近接リストを生成
      IDX = 0

      write(*,*) 'make_list'
      CALL HAKE_LIST(N,ZAHYO,LIST,KYORI,CUTOFF,IDX)

      write(*,*) 'make_list2'
C     第二近接リストを生成
      CALL HAKE_LIST2(N,LIST,LIST2)


      write(*,*) IDX
      write(*,*) 'make_list_vdw'
      CALL HAKE_LIST_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,IDX,LIST,
     #LIST2,AN)

      write(*,*) IDX
      write(*,*) 'make_newkyori'
      CALL HAKE_NEWKYORI(N,ZAHYO,LIST,KYORI,CUTOFF)

      write(*,*) 'make_newkyoriv'
      CALL HAKE_NEWKYORIV(N,ZAHYO,LIST_VDW,KYORI,CUTOFF)

      write(*,*) 'calc_diff'
      CALL CALC_DIFF1(N,ZAHYO,LIST,KYORI,CUTOFF,DIFF1)

      write(*,*) 'calc_diff_vdw'
      CALL CALC_DIFF1_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF1)


C      DO 38 J = 1 , HAX_ATOH3
C         DIFF2(0,J) = 0
C         DD(J) = 0.0D0
C 38   CONTINUE

C       CALL CALC_DIFF2(N,ZAHYO,LIST,LIST2,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
C       CALL CALC_DIFF2_VDW
C     #      (N,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF2,DIFF2_DATA)


C      CALL CONJGRAD(N,ZAHYO,DIFF1,DIFF2,DIFF2_DATA)

      DO 39 J = 1 , N*3
          VELO(J) = VELO(J) + DIFF1(J) * CONV_VELO
         ZAHYO(J) = ZAHYO(J) + VELO(J)
          VELO(J) = VELO(J) * 0.9D0

 39    CONTINUE


       ENERGY = TERSOFF(N,ZAHYO,LIST,KYORI,CUTOFF)
       ENERGY_VDW = VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF)
       ENERGY = ENERGY +  ENERGY_VDW

       WRITE(20,40) I , ENERGY

 40    format(i6,f20.7)



C       K = K + 1
C       IF (K .EQ. 10) THEN
C          CALL PRINT_ZAHYO(N,ZAHYO,I,ENERGY,DIFF1,AN)
C          K = 0
C       ENDIF
C IF ( I .EQ. 10 ) THEN
```

```
C CALL PRINT_ZAHYO(N,ZAHYO,I,ENERGY,DIFF1,AN)
C ENDIF

 41   CONTINUE
      CALL PRINT_ZAHYO(N,ZAHYO,I,ENERGY,DIFF1,AN)

      close(20)

      STOP
      END

c
c This program is for md to run on Dec Fortran
c

c      function IARGC()
c      IARGC=1
c      return
c      end
c      subroutine GETARG(i,c)
c      character*80 c
c      open(unit=53,file='jobname.dat',status='old')
c      read(53,45) c
c 45   format(a30)
c      close(53)
c      return
c      end
c      subroutine FDATE(IDATE)
c      CHARACTER IDATE*24
c      call DATE(IDATE)
c      return
c      end

CC
C     座標データの FILE 名を取得する サブルーチン
CC
 101  SUBROUTINE GETFILENAHE(FILENAHE)
      CHARACTER*50 FILENAHE
      INTEGER I
      I = IARGC()
      IF (I .NE. 1) then
         WRITE(*,*) 'Usage: opt hoge.xyz'
         GOTO 110
      ENDIF

      CALL GETARG(1,FILENAHE)

      RETURN
 110  STOP
      END


 121  SUBROUTINE GETARGU(LINE,I,ARGU)
      CHARACTER*80 LINE
      CHARACTER*80 DUH
      CHARACTER*80 ARGU
      INTEGER I
      INTEGER J
      INTEGER K
      INTEGER lenline

      DUH = LINE


      do 128 K = 1 , I
         lenline = len(DUH)
         DO 122 J = 1 , lenline

            IF ( DUH(J:J) .EQ. ' ') THEN
               GOTO 122
            ENDIF
            IF ( DUH(J:J) .EQ. char(9) ) THEN
               GOTO 122
            ENDIF
            IF ( DUH(J:J) .EQ. char(0) ) THEN
               GOTO 122
            ENDIF
            GOTO 123
 122     CONTINUE

 123     DUH = DUH(J:lenline)

         lenline = len(DUH)
         DO 124 J = 2 , lenline
            IF ( DUH(J:J) .EQ. ' ') THEN
               GOTO 125
            ENDIF
            IF ( DUH(J:J) .EQ. char(9) ) THEN
               GOTO 125
            ENDIF
            IF ( DUH(J:J) .EQ. char(0) ) THEN
               GOTO 125
```

```
           ENDIF
124     CONTINUE
125     ARGU = DUH(1:J-1)

        DUH=DUH(J:lenline)

128   continue

      RETURN

129   STOP
      END

130   REAL*8 FUNCTION ATOR(STR)
      CHARACTER*80 STR
      INTEGER I
      INTEGER J
      INTEGER S
      INTEGER D
      INTEGER DCU
      INTEGER*4 R
      INTEGER*4 E

      I = 1
      S = 1

      R = 0
      EC = 0
      E = 0

      IF ( STR(1:1) .EQ. '+') THEN
         S = 1
         I = I + 1
      ENDIF
      IF ( STR(1:1) .EQ. '-') THEN
         S = -1
         I = I + 1
      ENDIF

      DO 140 J = I , 80
         IF ( STR(J:J) .EQ. '.' ) THEN
            EC = 1
            GOTO 140
         ENDIF

         D = ICHAR(STR(J:J)) - ICHAR('0')

         IF ( (D .GT. 9) .OR. (D .LT. 0)) THEN
         GOTO 145
      ENDIF
      R = R * 10 + D
      E = E + EC

      IF ( R .GT. 9999999 ) THEN
         GOTO 145
      ENDIF

140   CONTINUE

145   ATOR = 1.0D0  * S * R / (10.0D0**E)

      RETURN

149   STOP
      END



CC
C     座標データ取り込み サブルーチン
CC

201   SUBROUTINE READ_ZAHYO(FILENAHE, N, ZAHYO,AN)
      INCLUDE 'PARAHETER'
      CHARACTER*50 FILENAHE

C     原子数を保持する変数
      INTEGER N
      INTEGER N3

C     FILE IO チェック用変数
      INTEGER IOCHECK

C     loop variable
      INTEGER I

C     炭素原子の座標用配列
      REAL*8 ZAHYO(HAX_ATOH3)

C     元素名用変数
      CHARACTER*8 AN(HAX_ATOH)
```

```
      CHARACTER*80 LINE,ARGU

C     FILE OEPN
      OPEN(60,FILE=FILENAME,STATUS='OLD',IOSTAT=IOCHECK)

C     FILE OPEN のエラーチェック
      IF ( IOCHECK ) THEN
         WRITE(*,*) 'File open error.'
         GOTO 299
      ENDIF

C     原子数の制限チェック
      READ(60,*) N

      IF ( MAX_ATOM .LT. N ) THEN
         WRITE(*,*) "Too many atoms."
         GOTO 299
      ENDIF

      N3 = N * 3

      READ(60,209) LINE

 208  format(f14.8)
 209  format(a80)

C     座標読み込み 単位はÅ
      DO 210 I = 1 , N3 , 3
         READ(60,209) LINE
         CALL GETARGU(LINE,1,ARGU)
         AN(I/3+1)=ARGU
         CALL GETARGU(LINE,2,ARGU)
         ZAHYO(I) = ATOR(ARGU)
         CALL GETARGU(LINE,3,ARGU)
         ZAHYO(I+1) = ATOR(ARGU)
         CALL GETARGU(LINE,4,ARGU)
         ZAHYO(I+2) = ATOR(ARGU)

C         READ(60,*) AN(I/3+1) , ZAHYO(I) ,ZAHYO(I+1) ,ZAHYO(I+2)
 210  CONTINUE

      CLOSE(60,STATUS='KEEP')
      RETURN

 299  CLOSE(60,STATUS='KEEP')
      STOP
      END

 301  SUBROUTINE PRINT_ZAHYO(N,ZAHYO,T,ENERGY,DIFF1,AN)
      INCLUDE 'PARAMETER'
      INTEGER N,N3
      INTEGER T
      REAL*8 ZAHYO(MAX_ATOM3)
      CHARACTER*8 AN(MAX_ATOM)
      REAL*8 DIFF1(MAX_ATOM3)
      REAL*8 ENERGY
      INTEGER I

      WRITE(*,*) N
      WRITE(*,*) 'TIME=',T*TIME_DIV*1.0D15,'fs  ENERGY =',ENERGY

      N3 = N * 3

      DO 311 I = 1 , N3 , 3
         WRITE(*,305) AN(I/3+1) , ZAHYO(I) , ZAHYO(I+1) , ZAHYO(I+2)
     #    , DIFF1(I) *10.0 , DIFF1(I+1) * 10.0 , DIFF1(I+2) * 10.0

 305     FORMAT(a5,f12.7,f12.7,f12.7,f12.7,f12.7,f12.7)
 311  CONTINUE

 398  RETURN
 399  STOP
      END

CC
C     最近接原子の LIST を作成するサブルーチン
CC

C
C     LIST(0,N) : 原子番号 N の近接原子の数
C     LIST(I,N) : 原子番号 N の I 番目の近接原子インデックス
C     LIST(I+MAX_LIST_N ,N) : 原子番号 N の I 番目の近接原子番号

 401  SUBROUTINE MAKE_LIST(N,ZAHYO,LIST,KYORI,CUTOFF,IDX)
      INCLUDE 'PARAMETER'
      INTEGER N,N3

      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)
      INTEGER IDX
```

```
      REAL*8  MAX_X,MAX_Y,MAX_Z
      REAL*8  MIN_X,MIN_Y,MIN_Z

      INTEGER I,K,J
      INTEGER E,F,G
      INTEGER L,H

      INTEGER I3

      INTEGER MESH_X,MESH_Y,MESH_Z
      INTEGER TX,TY,TZ
      INTEGER TI,TJ,TI3,TJ3

      INTEGER MESH(0:MAX_MESH_ATOM,
     #       0:MAX_MESH_X+1,0:MAX_MESH_Y+1,0:MAX_MESH_Z+1)

      REAL*8 THP_R,THP_CUTOFF

      REAL*8 SIN

      N3 = N * 3

C     リスト配列の初期化
      DO 405 I = 1 , MAX_ATOM
         LIST(0,I) = 0
 405  CONTINUE

C     MESH 配列の初期化
      DO 406 I = 0 , MAX_MESH_X+1
         DO 406 J = 0 , MAX_MESH_Y+1
            DO 406 K = 0 , MAX_MESH_Z+1
               MESH(0,I,J,K) = 0
 406  CONTINUE

C     系の座標の最大値 - 最小値を検索する
      MAX_X = ZAHYO(1)
      MAX_Y = ZAHYO(2)
      MAX_Z = ZAHYO(3)
      MIN_X = ZAHYO(1)
      MIN_Y = ZAHYO(2)
      MIN_Z = ZAHYO(3)

      DO 410 I = 4 , N3 , 3

         IF ( MAX_X .LT. ZAHYO(I) ) THEN
            MAX_X = ZAHYO(I)
         ENDIF
         IF ( MAX_Y .LT. ZAHYO(I+1) ) THEN
            MAX_Y = ZAHYO(I+1)
         ENDIF
         IF ( MAX_Z .LT. ZAHYO(I+2) ) THEN
            MAX_Z = ZAHYO(I+2)
         ENDIF
         IF ( MIN_X .GT. ZAHYO(I) ) THEN
            MIN_X = ZAHYO(I)
         ENDIF
         IF ( MIN_Y .GT. ZAHYO(I+1) ) THEN
            MIN_Y = ZAHYO(I+1)
         ENDIF
         IF ( MIN_Z .GT. ZAHYO(I+2) ) THEN
            MIN_Z = ZAHYO(I+2)
         ENDIF
 410  CONTINUE

C     一度、大まかに原子をグルーピングする。
C     最大値最小値から、メッシュの個数を決める

      MESH_X = 1 + ( ( MAX_X - MIN_X ) / MESH_D )
      MESH_Y = 1 + ( ( MAX_Y - MIN_Y ) / MESH_D )
      MESH_Z = 1 + ( ( MAX_Z - MIN_Z ) / MESH_D )

C     MESH 用の配列 に収まらない場合 STOP
      IF ( MESH_X .LE. MAX_MESH_X ) THEN
         GOTO 415
      ENDIF
      IF ( MESH_Y .LE. MAX_MESH_Y ) THEN
         GOTO 415
      ENDIF
      IF ( MESH_Z .LE. MAX_MESH_Z ) THEN
         GOTO 415
      ENDIF

      WRITE(*,*) 'MESH is overflow: MAX_MESH.',  MESH_X,MESH_Y,MESH_Z
      STOP


C     全ての原子を MESH に グルーピングする。

 415  DO 420 I = 1 , N
         I3 = I * 3 - 2
         TX = 1 + ( ( ZAHYO(I3) - MIN_X ) / MESH_D )
```

```
         TY = 1 + ( ( ZAHYO(I3+1) - MIN_Y ) / HESH_D )
         TZ = 1 + ( ( ZAHYO(I3+2) - MIN_Z ) / HESH_D )

C     HESH 配列があるれる場合 STOP
         IF ( HESH(0,TX,TY,TZ) .EQ. MAX_HESH_ATOH ) THEN
            WRITE(*,*) 'HESH_ATOH is overflow: MAX_HESH_ATOH.'
            STOP
         ENDIF

         HESH(0,TX,TY,TZ) =  HESH(0,TX,TY,TZ) + 1
         HESH( HESH(0,TX,TY,TZ) , TX,TY,TZ) = I

 420  CONTINUE


C     HESH のブロックの 近接内 (27 ブロック)
C     で 距離を計算し 近接LIST を作成する
C     ついでに カットオフ関数の値も計算する

      DO 443 I = 1 , HESH_X
         DO 442 J = 1 , HESH_Y
            DO 441 K = 1 , HESH_Z

               IF ( HESH(0,I,J,K) .EQ. 0 ) THEN
                  GOTO 441
               ENDIF

               DO 433 E = I - 1  , I + 1
                  DO 432 F = J - 1 ,  J + 1
                     DO 431 G = K - 1  , K + 1

                        IF ( HESH(0,E,F,G) .EQ. 0 ) THEN
                           GOTO 431
                        ENDIF

                        DO 425 L = 1 , HESH(0,I,J,K)
                           DO 424 H = 1 , HESH(0,E,F,G)

                              TI = HESH(L,I,J,K)
                              TJ = HESH(H,E,F,G)

                              IF ( TI .LE. TJ ) THEN
                                 GOTO 424
                              ENDIF

                              TI3 = TI*3-2
                              TJ3 = TJ*3-2
C
C     原子間距離の計算
C
      THP_R =
     #     ( ( ZAHYO(TJ3  ) - ZAHYO(TI3  ) )**2 +
     #       ( ZAHYO(TJ3+1) - ZAHYO(TI3+1) )**2 +
     #       ( ZAHYO(TJ3+2) - ZAHYO(TI3+2) )**2  )**0.5d0

C
C     近接でない場合場合 処理を飛ばす
C
      IF ( THP_R .GT. (PRH_CR + PRH_CD) ) THEN
         GOTO 424
      ENDIF

C     LIST の配列があふれる時 STOP

      IF ( ( LIST(0,TI) .EQ. MAX_LIST_N ) .OR.
     #     ( LIST(0,TJ) .EQ. MAX_LIST_N ) ) THEN
         WRITE(*,*) 'LIST_N is overflow: MAX_LIST_N.'
         STOP
      ENDIF

      IDX = IDX + 1
      KYORI(IDX) = THP_R

      LIST(0,TI) = LIST(0,TI) + 1
      LIST( LIST(0,TI) ,TI ) = IDX
      LIST( LIST(0,TI) + MAX_LIST_N ,TI ) = TJ

      LIST(0,TJ) = LIST(0,TJ) + 1
      LIST( LIST(0,TJ) ,TJ ) = IDX
      LIST( LIST(0,TJ) + MAX_LIST_N ,TJ ) = TI

C
C     カットオフ関数を計算
C

      THP_CUTOFF = 1.0D0
      IF ( THP_R .GE. ( PRH_CR - PRH_CD) ) THEN
         THP_CUTOFF = 0.5D0 *
     #   ( 1.0D0 - SIN( PI * ( THP_R - PRH_CR ) / (2.0D0 * PRH_CD ) ))
      ENDIF
      IF ( THP_R .GE. ( PRH_CR + PRH_CD) ) THEN
         THP_CUTOFF = 0.0D0
```

```
      ENDIF

      CUTOFF(IDX) = THP_CUTOFF

 424  CONTINUE
 425  CONTINUE


 431  CONTINUE
 432  CONTINUE
 433  CONTINUE

 441  CONTINUE
 442  CONTINUE
 443  CONTINUE

C      DO 452 I = 1 , N
C         write(*,*) "--",I
C         DO 451 J = 1 , LIST(0,I)
C            WRITE(*,*) LIST(J + MAX_LIST_N , I)
C 451     CONTINUE
C 452  CONTINUE

 469  RETURN
 499  STOP
      END

C
C     i 原子に関する Tersoff ポテンシャル
C

 501  REAL*8 FUNCTION TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
      INCLUDE 'PARAMETER'
      INTEGER I

      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER J,K,L,M
      INTEGER I3,J3,K3
      INTEGER IDXJ,IDXK

      REAL*8 G,GCOS
      REAL*8 ZETA
      REAL*8 I_X,I_Y,I_Z
      REAL*8 J_X,J_Y,J_Z
      REAL*8 K_X,K_Y,K_Z
      REAL*8 Bij

      REAL*8 EXP

      REAL*8 THP

      I3 = I*3 - 2
      I_X = ZAHYO(I3)
      I_Y = ZAHYO(I3+1)
      I_Z = ZAHYO(I3+2)

      THP = 0.0D0

      DO 520 L = 1 , LIST(0,I)
         IDXJ = LIST(L,I)
         J = LIST(L+MAX_LIST_N,I)
         J3 = J*3 - 2
         J_X = ZAHYO(J3)
         J_Y = ZAHYO(J3+1)
         J_Z = ZAHYO(J3+2)

         ZETA = 0.0D0

         DO 510 M = 1 , LIST(0,I)
            IF ( L .EQ. M ) THEN
               GOTO 510
            ENDIF

            IDXK = LIST(M,I)
            K = LIST(M+MAX_LIST_N,I)
            K3 = K*3 - 2
            K_X = ZAHYO(K3)
            K_Y = ZAHYO(K3+1)
            K_Z = ZAHYO(K3+2)

            GCOS =
     #           (
     #           (J_X - I_X ) * (K_X - I_X )+
     #           (J_Y - I_Y ) * (K_Y - I_Y )+
     #           (J_Z - I_Z ) * (K_Z - I_Z )
     #           ) / KYORI(IDXJ) / KYORI(IDXK)

            IF ( KYORI(IDXK) .EQ. 0.0D0 ) THEN
```

```
            WRITE(*,*) KYORI(IDXK) ,I,J,K
          ENDIF


          G = ( PRH_C * PRH_C ) / ( PRH_D * PRH_D ) -
     #           ( PRH_C * PRH_C ) /
     #           ( PRH_D * PRH_D + ( PRH_H - GCOS ) * (PRH_H - GCOS ))

          G = PRH_A * ( G + 1.0D0 )
          ZETA = ZETA + CUTOFF(IDXK) * G

510       CONTINUE

          Bij = ( 1.0 + ( ZETA )**( PRH_ETA) )**( -PRH_DELTA )

          THP = THP +
     #    CUTOFF(IDXJ) *
     #    (
     #     PRH_EA * EXP( -PRH_LUHBDA1 * KYORI(IDXJ) ) +
     #     (PRH_EB) * Bij * EXP( -PRH_LUHBDA2 * KYORI(IDXJ))
     #    )

520   CONTINUE

      TERSOFF_I = THP

      RETURN
599   STOP
      END

601   REAL*8 FUNCTION TERSOFF(N,ZAHYO,LIST,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I
      INTEGER N
      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)
      REAL*8  TERSOFF_I

      TERSOFF = 0.0D0

      DO 610 I = 1 , N
         TERSOFF = TERSOFF +
     #        TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)

610   CONTINUE

      TERSOFF = TERSOFF / 2.0D0

      RETURN
699   STOP
      END

701   SUBROUTINE CALC_DIFF1(N,ZAHYO,LIST,KYORI,CUTOFF,DIFF1)
      INCLUDE 'PARAHETER'
      INTEGER N,N3
      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      REAL*8  DIFF1(MAX_ATOM3)

      INTEGER I,ID3
      REAL*8  ZAHYO_ORG
      REAL*8 TERSOFF_LI

      REAL*8 THP

      N3 = N * 3

      DO 710 I = 1 , N3
         ID3 = ( I + 2 ) / 3
         ZAHYO_ORG = ZAHYO(I)

         ZAHYO(I) = ZAHYO_ORG + EPS1
         CALL RECALC(ID3,ZAHYO,LIST,KYORI,CUTOFF)
         THP = TERSOFF_LI(ID3,ZAHYO,LIST,KYORI,CUTOFF)

         ZAHYO(I) = ZAHYO_ORG - EPS1
         CALL RECALC(ID3,ZAHYO,LIST,KYORI,CUTOFF)
         THP = THP -
     #        TERSOFF_LI(ID3,ZAHYO,LIST,KYORI,CUTOFF)

         ZAHYO(I) = ZAHYO_ORG
         CALL RECALC(ID3,ZAHYO,LIST,KYORI,CUTOFF)

         THP = THP / ( 2.0 * EPS1 )

         DIFF1(I) = -THP
```

```
710  CONTINUE

     RETURN
799  STOP
     END

801  SUBROUTINE RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
     INCLUDE 'PARAHETER'
     INTEGER I,I3
     REAL*8  ZAHYO(HAX_ATOH3)
     INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
     REAL*8  KYORI(HAX_DATA_IDX)
     REAL*8  CUTOFF(HAX_DATA_IDX)

     INTEGER J,J3
     INTEGER IDXJ
     INTEGER H

     REAL*8 THP_R
     REAL*8 THP_CUTOFF

     REAL*8 SIN
     REAL*8 SQRT

     I3 = I*3-2

     DO 810 H = 1 , LIST(0,I)
        IDXJ = LIST(H,I)
        J = LIST(H+HAX_LIST_N,I)
        IDXJ = LIST(H,I)

        J3 = J*3-2
         THP_R =
    #    SQRT    ( ( ZAHYO(J3) - ZAHYO(I3) )**2 +
    #         ( ZAHYO(J3+1) - ZAHYO(I3+1) )**2 +
    #         ( ZAHYO(J3+2) - ZAHYO(I3+2) )**2  )


        KYORI(IDXJ) = THP_R

        THP_CUTOFF = 1.0D0
        IF ( THP_R .GE. ( PRH_CR - PRH_CD) ) THEN
        THP_CUTOFF = 0.5D0 *
    #   ( 1.0D0 - SIN( PI * ( THP_R - PRH_CR ) / (2.0D0 * PRH_CD ) ))
        ENDIF
        IF ( THP_R .GE. ( PRH_CR + PRH_CD) ) THEN
           THP_CUTOFF = 0.0D0
        ENDIF

        CUTOFF(IDXJ) = THP_CUTOFF

810  CONTINUE

     RETURN
899  STOP
     END

901  REAL*8 FUNCTION TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)
     INCLUDE 'PARAHETER'
     INTEGER I
     REAL*8  ZAHYO(HAX_ATOH3)
     INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
     REAL*8  KYORI(HAX_DATA_IDX)
     REAL*8  CUTOFF(HAX_DATA_IDX)
     REAL*8  TERSOFF_I
     INTEGER L

     INTEGER J

     TERSOFF_LI = TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)

     DO 910 L = 1 , LIST(0,I)
        J = LIST(L + HAX_LIST_N,I)
        TERSOFF_LI = TERSOFF_LI +
    #        TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)

910  CONTINUE

     RETURN

999  STOP
     END


1001 SUBROUTINE HAKE_LIST2(N,LIST,LIST2)
     INCLUDE 'PARAHETER'
     INTEGER N
     INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
     INTEGER LIST2(0:HAX_LIST2_N,HAX_ATOH)

     INTEGER I,J,K,L,H
```

```
      DO 1010 I = 1 , N
         LIST2(0,I) = 0

         DO 1005 J = 1 , LIST(0,I)
            L = LIST(J+HAX_LIST_N,I)

            DO 1003 K = 1 , LIST(0,L)
               H = LIST(K+HAX_LIST_N,L)
               IF ( H .EQ. I) THEN
                  GOTO 1003
               ENDIF
               IF ( H .EQ. L) THEN
                  GOTO 1003
               ENDIF

               IF ( LIST2(0,I) .EQ. HAX_LIST2_N ) THEN
                  WRITE(*,*) 'HAX_LIST2_N is overflow'
                  STOP
               ENDIF

               LIST2(0,I) = LIST2(0,I) + 1
               LIST2(LIST2(0,I),I) = H

 1003       CONTINUE
 1005    CONTINUE
 1010 CONTINUE

      RETURN
 1099 STOP
      END

 1101 SUBROUTINE CALC_DIFF2
     #(N,ZAHYO,LIST,LIST2,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
      INCLUDE 'PARAHETER'

      INTEGER N,N3
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      INTEGER LIST2(0:HAX_LIST2_N,HAX_ATOH)

      REAL*8  ZAHYO(HAX_ATOH3)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)

      INTEGER I,J,K
      INTEGER L

      N3 = N * 3

C      DO 1110 I = 1 , HAX_ATOH3
C         DIFF2(0,I) = 0
C 1110 CONTINUE

      DO 1150 I = 1 , N
C     同一原子の座標系
         CALL CALC_DIFF2_1
     #        (I,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

C     隣合う原子の座標系
         DO 1145 L = 1 , LIST(0,I)
            J = LIST(L+HAX_LIST_N,I)

            CALL CALC_DIFF2_2
     #           (I,J,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
 1145    CONTINUE


C     第二隣接の座標系

         DO 1146 L = 1 , LIST2(0,I)
            J = LIST2(L,I)
            CALL CALC_DIFF2_3
     #           (I,J,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
 1146       CONTINUE

 1150 CONTINUE

      RETURN
 1199 STOP
      END

 1201 SUBROUTINE CALC_DIFF2_1
     #     (I,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

      INCLUDE 'PARAHETER'
      INTEGER I
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)

      REAL*8  ZAHYO(HAX_ATOH3)
      REAL*8  KYORI(HAX_DATA_IDX)
```

```
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  TERSOFF_LI

      INTEGER I3 , J3
      INTEGER II,JJ
      INTEGER I3II,J3JJ
      REAL*8 X,Y

      REAL*8 THP
      REAL*8 DATA

      I3 = I*3 - 3
      J3 = I3

      DO 1230 II = 1 , 3
         DO 1220 JJ = 1 , 3
            I3II = I3 + II
            J3JJ = J3 + JJ

            IF ( I3II .NE. J3JJ ) THEN

               X = ZAHYO(I3II)
               Y = ZAHYO(J3JJ)

               ZAHYO(I3II) = X + EPS2
               ZAHYO(J3JJ) = Y + EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X - EPS2
               ZAHYO(J3JJ) = Y - EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP +
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X + EPS2
               ZAHYO(J3JJ) = Y - EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP -
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X - EPS2
               ZAHYO(J3JJ) = Y + EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP -
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               DATA = THP / ( 8.0D0 * EPS2 * EPS2 )

               ZAHYO(I3II) = X
               ZAHYO(J3JJ) = Y

            ELSE

               X = ZAHYO(I3II)

               ZAHYO(I3II) = X + 2.0D0*EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X - 2.0D0*EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP +
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X + EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP -
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X - EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP -
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               DATA = THP / ( 3.0D0 * EPS2 * EPS2 )

               ZAHYO(I3II) = X

            ENDIF

            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL ADD_DIFF2( DATA , I3II , J3JJ , DIFF2 ,DIFF2_DATA)

 1220    CONTINUE
 1230 CONTINUE

      RETURN
 1299 STOP
```

```
      END

 1301 SUBROUTINE ADD_DIFF2( DATA , I3II , J3JJ , DIFF2 , DIFF2_DATA )
      INCLUDE 'PARAHETER'

      INTEGER I3II , J3JJ
      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DATA

      IF ( DIFF2(0,I3II) .EQ. HAX_DIFF2_LIST ) THEN
         WRITE(*,*) 'HAX_DIFF2_LIST is overflow.'
         STOP
      ENDIF

      DIFF2(0,I3II) = DIFF2(0,I3II) + 1
      DIFF2(DIFF2(0,I3II),I3II) = J3JJ
      DIFF2_DATA(DIFF2(0,I3II),I3II) = DATA
C      WRITE(*,*) I3II , J3JJ, DATA

      RETURN
 1349 STOP
      END

 1351 SUBROUTINE ADD2_DIFF2( DATA , I3II , J3JJ , DIFF2 , DIFF2_DATA )
      INCLUDE 'PARAHETER'

      INTEGER I3II , J3JJ
      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DATA
      INTEGER I


      DO 1360 I = 1 , DIFF2(0,I3II)
         IF ( DIFF2(I,I3II) .EQ. J3JJ ) THEN
            DIFF2_DATA(I,I3II) = DIFF2_DATA(I,I3II) + DATA
            GOTO 1380
         ENDIF
 1360 CONTINUE

      WRITE(*,*) 'error ADD2_DIFF2'
      STOP
C     WRITE(*,*) I3II , J3JJ, DATA



 1380 RETURN
 1399 STOP
      END




 1401 SUBROUTINE CALC_DIFF2_2
     #     (I,J,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

      INCLUDE 'PARAHETER'
      INTEGER I,J
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      INTEGER LIST_IJ(0:HAX_LIST_IJ)

      REAL*8  ZAHYO(HAX_ATOH3)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  TERSOFF_I
      REAL*8  TERSOFF_LIJ

      INTEGER I3 , J3
      INTEGER II,JJ
      INTEGER I3II,J3JJ
      REAL*8 X,Y

      REAL*8 THP
      REAL*8 DATA

      CALL HAKE_LIST_IJ( I,J,LIST_IJ,LIST )

      I3 = I*3 - 3
      J3 = J*3 - 3

      DO 1430 II = 1 , 3
         DO 1420 JJ = 1 , 3
            I3II = I3 + II
            J3JJ = J3 + JJ

               X = ZAHYO(I3II)
               Y = ZAHYO(J3JJ)
```

```
              ZAHYO(I3II) = X + EPS2
              ZAHYO(J3JJ) = Y + EPS2
              CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
              CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)
              THP = TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
     #              +TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)
     #              +TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

              ZAHYO(I3II) = X - EPS2
              ZAHYO(J3JJ) = Y - EPS2
              CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
              CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

              THP = THP
     #              +TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
     #              +TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)
     #              +TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

              ZAHYO(I3II) = X + EPS2
              ZAHYO(J3JJ) = Y - EPS2

              CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
              CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

              THP = THP
     #              -TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
     #              -TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)
     #              -TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

              ZAHYO(I3II) = X - EPS2
              ZAHYO(J3JJ) = Y + EPS2
              CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
              CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

              THP = THP
     #              -TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
     #              -TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)
     #              -TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

              DATA = THP / ( 8.0D0 * EPS2 * EPS2 )

              ZAHYO(I3II) = X
              ZAHYO(J3JJ) = Y

              CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
              CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

              CALL ADD_DIFF2( DATA , I3II , J3JJ , DIFF2 ,DIFF2_DATA)


1420     CONTINUE
1430  CONTINUE

      RETURN
1499  STOP
      END

1501  SUBROUTINE HAKE_LIST_IJ( I,J,LIST_IJ,LIST )
      INCLUDE 'PARAHETER'

      INTEGER I,J
      INTEGER LIST_IJ(0:HAX_LIST_IJ)
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)

      INTEGER L,H

      LIST_IJ(0) = 0

      DO 1540 L = 1 , LIST(0,I)
         DO 1530 H = 1 , LIST(0,J)
            IF ( LIST(L+HAX_LIST_N,I)
     #           .NE. LIST(H+HAX_LIST_N,J)) THEN
               GOTO 1530
            ENDIF

            IF ( LIST_IJ(0) .EQ. HAX_LIST_IJ )  THEN
               WRITE(*,*) 'HAX_LIST_IJ is overflow.'
            ENDIF

            LIST_IJ(0) = LIST_IJ(0) + 1
            LIST_IJ(LIST_IJ(0)) = LIST(L+HAX_LIST_N,I)

1530     CONTINUE
1540  CONTINUE

      RETURN
1599  STOP
      END

1601  REAL*8 FUNCTION TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER LIST_IJ(0:HAX_LIST_IJ)
```

```
      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)
      REAL*8  TERSOFF_I

      INTEGER I,L

      TERSOFF_LIJ = 0.0D0

      DO 1610 L = 1 , LIST_IJ(0)

         I = LIST_IJ(L)

         TERSOFF_LIJ = TERSOFF_LIJ +
     #        TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)

1610  CONTINUE

      RETURN

1699  STOP
      END

1701 SUBROUTINE CALC_DIFF2_3
     #     (I,J,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

      INCLUDE 'PARAMETER'
      INTEGER I,J
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      INTEGER LIST_IJ(0:MAX_LIST_IJ)

      REAL*8  ZAHYO(MAX_ATOM3)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  TERSOFF_LIJ

      INTEGER I3 , J3
      INTEGER II,JJ
      INTEGER I3II,J3JJ
      REAL*8 X,Y

      REAL*8 THP
      REAL*8 DATA

      CALL MAKE_LIST_IJ( I,J,LIST_IJ,LIST )

      I3 = I*3 - 3
      J3 = J*3 - 3

      DO 1730 II = 1 , 3
         DO 1720 JJ = 1 , 3
            I3II = I3 + II
            J3JJ = J3 + JJ

            X = ZAHYO(I3II)
            Y = ZAHYO(J3JJ)

            ZAHYO(I3II) = X + EPS2
            ZAHYO(J3JJ) = Y + EPS2
            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)


            THP = TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

            ZAHYO(I3II) = X - EPS2
            ZAHYO(J3JJ) = Y - EPS2
            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

            THP = THP
     #           +TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

            ZAHYO(I3II) = X + EPS2
            ZAHYO(J3JJ) = Y - EPS2
            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

            THP = THP
     #           -TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

            ZAHYO(I3II) = X - EPS2
            ZAHYO(J3JJ) = Y + EPS2
            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

            THP = THP
     #           -TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)
```

```
            DATA = THP / ( 8.0D0 * EPS2 * EPS2 )

            ZAHYO(I3II) = X
            ZAHYO(J3JJ) = Y

            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

            CALL ADD_DIFF2( DATA , I3II , J3JJ , DIFF2 ,DIFF2_DATA)

 1720    CONTINUE
 1730 CONTINUE

      RETURN
 1799 STOP
      END

 1801 SUBROUTINE CONJGRAD(N,ZAHYO,DIFF1,DIFF2,DIFF2_DATA)
      INCLUDE 'PARAHETER'
      INTEGER N
      REAL*8 ZAHYO(MAX_ATOM3)
      REAL*8 DIFF1(MAX_ATOM3)
      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)

      INTEGER N3

      REAL*8 P(MAX_ATOM3)
      REAL*8 R(MAX_ATOM3)
      REAL*8 AP(MAX_ATOM3)
      REAL*8 X(MAX_ATOM3)

      REAL*8 NORM_R2,NORM_R2_

      REAL*8 PAP

      REAL*8 A

      REAL*8 C

      INTEGER I,J,K
      INTEGER L

      N3 = N*3
      NORM_R2 = 0.0D0

      DO 1810 I = 1 , N3
         X(I) = 0.0D0
         R(I) = DIFF1(I)
         P(I) = DIFF1(I)
         NORM_R2 = NORM_R2 + R(I) * R(I)
 1810 CONTINUE

C      WRITE(*,*) 0,NORM_R2
      DO 1860 K = 1 , N3

         IF ( NORM_R2 .LT. 1.0D-6 ) THEN
            GOTO 1865
         ENDIF

         DO 1820 I = 1 , N3
            AP(I) = 0.0D0
            DO 1815 L = 1 , DIFF2(0,I)
               J = DIFF2(L,I)
               AP(I) = AP(I) + P(J) * DIFF2_DATA(L,I)

 1815       CONTINUE
 1820    CONTINUE

         PAP = 0.0D0
         DO 1830 I = 1 , N3
            PAP = PAP + P(I) * AP(I)
 1830    CONTINUE

         A = NORM_R2 / PAP
C         WRITE(*,*) 'A=',A

         NORM_R2_ = 0.0D0

         DO 1840 I = 1 , N3
            X(I) = X(I) + A * P(I)
            R(I) = R(I) - A * AP(I)
            NORM_R2_ = NORM_R2_ + R(I) * R(I)
 1840    CONTINUE

C         WRITE(*,*) K,NORM_R2_

         C = NORM_R2_ / NORM_R2
         IF ( C .GT. 1.0D0 ) THEN
            GOTO 1865
         ENDIF
```

```
         NORH_R2 = NORH_R2_
         DO 1850 I = 1 , N3
            P(I) = R(I) + C * P(I)
 1850    CONTINUE

 1860 CONTINUE

 1865 DO 1870 I = 1 , N3
         ZAHYO(I) = ZAHYO(I) + X(I)
 1870 CONTINUE

      RETURN
 1899 STOP
      END

 1901 SUBROUTINE HAKE_LIST_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,IDX
     #,LIST,LIST2,AN)
      INCLUDE 'PARAHETER'
      INTEGER N,N3

      REAL*8  ZAHYO(HAX_ATOH3)
      CHARACTER*8  AN(HAX_ATOH)
      INTEGER LIST_VDW(0:HAX_LIST_VN2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)
      INTEGER IDX

      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      INTEGER LIST2(0:HAX_LIST2_N,HAX_ATOH)

      REAL*8  HAX_X,HAX_Y,HAX_Z
      REAL*8  HIN_X,HIN_Y,HIN_Z

      INTEGER I,K,J
      INTEGER E,F,G
      INTEGER L,H

      INTEGER I3

      INTEGER HESH_X,HESH_Y,HESH_Z
      INTEGER TX,TY,TZ
      INTEGER TI,TJ,TI3,TJ3

      INTEGER HESH_V(0:HAX_HESH_V_ATOH,
     #    0:HAX_HESH_V_X+1,0:HAX_HESH_V_Y+1,0:HAX_HESH_V_Z+1)


      REAL*8 THP_R,THP_CUTOFF

C      REAL*8 SIN

      INTEGER IDX

      N3 = N * 3

C     リスト配列の初期化
      DO 1905 I = 1 , HAX_ATOH
         LIST_VDW(0,I) = 0
 1905  CONTINUE

C     HESH 配列の初期化
      DO 1906 I = 0 , HAX_HESH_V_X+1
         DO 1906 J = 0 , HAX_HESH__V_Y+1
            DO 1906 K = 0 , HAX_HESH_V_Z+1
               HESH_V(0,I,J,K) = 0
 1906  CONTINUE

C     系の座標の最大値 - 最小値を検索する

      HAX_X = ZAHYO(1)
      HAX_Y = ZAHYO(2)
      HAX_Z = ZAHYO(3)
      HIN_X = ZAHYO(1)
      HIN_Y = ZAHYO(2)
      HIN_Z = ZAHYO(3)

      DO 1910 I = 4 , N3 , 3

         IF ( HAX_X .LT. ZAHYO(I) ) THEN
            HAX_X = ZAHYO(I)
         ENDIF
         IF ( HAX_Y .LT. ZAHYO(I+1) ) THEN
            HAX_Y = ZAHYO(I+1)
         ENDIF
         IF ( HAX_Z .LT. ZAHYO(I+2) ) THEN
            HAX_Z = ZAHYO(I+2)
         ENDIF
         IF ( HIN_X .GT. ZAHYO(I) ) THEN
            HIN_X = ZAHYO(I)
         ENDIF
```

```
         IF ( HIN_Y .GT. ZAHYO(I+1) ) THEN
            HIN_Y = ZAHYO(I+1)
         ENDIF
         IF ( HIN_Z .GT. ZAHYO(I+2) ) THEN
            HIN_Z = ZAHYO(I+2)
         ENDIF
 1910  CONTINUE

C     一度、大まかに原子をグルーピングする。
C     最大値最小値から、メッシュの個数を決める


      HESH_X = 1 + ( ( HAX_X - HIN_X ) / HESH_D_VDW )
      HESH_Y = 1 + ( ( HAX_Y - HIN_Y ) / HESH_D_VDW )
      HESH_Z = 1 + ( ( HAX_Z - HIN_Z ) / HESH_D_VDW )

C     HESH 用の配列 に収まらない場合 STOP
      IF ( HESH_X .LE. HAX_HESH_V_X ) THEN
         GOTO 1915
      ENDIF
      IF ( HESH_Y .LE. HAX_HESH_V_Y ) THEN
         GOTO 1915
      ENDIF
      IF ( HESH_Z .LE. HAX_HESH_V_Z ) THEN
         GOTO 1915
      ENDIF

      WRITE(*,*) 'HESH is overflow-vdw: HAX_V_HESH.',  HESH_X,HESH_Y,HESH_Z
      STOP



C     全ての原子を HESH に グルーピングする。

 1915  DO 1920 I = 1 , N
         I3 = I * 3 - 2
         TX = 1 + ( ( ZAHYO(I3) - HIN_X ) / HESH_D_VDW )
         TY = 1 + ( ( ZAHYO(I3+1) - HIN_Y ) / HESH_D_VDW )
         TZ = 1 + ( ( ZAHYO(I3+2) - HIN_Z ) / HESH_D_VDW )


C     HESH 配列があるれる場合 STOP
         IF ( HESH_V(0,TX,TY,TZ) .GE. HAX_HESH_V_ATOH ) THEN
            WRITE(*,*) 'HESH_V_ATOH is overflow: HAX_HESH_V_ATOH.'
            STOP
         ENDIF

         HESH_V(0,TX,TY,TZ) =  HESH_V(0,TX,TY,TZ) + 1
         HESH_V( HESH_V(0,TX,TY,TZ) , TX,TY,TZ) = I

 1920  CONTINUE

C     HESH のブロックの 近接内 (27 ブロック)
C     で 距離を計算し 近接LIST を作成する
C     ついでに カットオフ関数の値も計算する

      DO 1943 I = 1 , HESH_X
         DO 1942 J = 1 , HESH_Y
            DO 1941 K = 1 , HESH_Z

               IF ( HESH_V(0,I,J,K) .EQ. 0 ) THEN
                  GOTO 1941
               ENDIF

               DO 1933 E = I - 1  , I + 1
                  DO 1932 F = J - 1 ,  J + 1
                     DO 1931 G = K - 1  , K + 1


                        DO 1925 L = 1 , HESH_V(0,I,J,K)
                            DO 1924 H = 1 , HESH_V(0,E,F,G)

                               TI = HESH_V(L,I,J,K)
                               TJ = HESH_V(H,E,F,G)

                               IF ( TI .LE. TJ ) THEN
                                  GOTO 1924
                               ENDIF

                               TI3 = TI*3-2
                               TJ3 = TJ*3-2

C
C     原子間距離の計算
C
      THP_R =
     #     ( ( ZAHYO(TJ3  ) - ZAHYO(TI3  ) )**2 +
     #       ( ZAHYO(TJ3+1) - ZAHYO(TI3+1) )**2 +
     #       ( ZAHYO(TJ3+2) - ZAHYO(TI3+2) )**2  )**0.5d0

C
C     範囲外のとき飛ばす。
```

```
      IF ( THP_R .GT. (PRH_VCR2 + PRH_VCD2) ) THEN
      GOTO 1924
      ENDIF

C     同一分子内は 無視
      IF ( AN(TI) .EQ.  AN(TJ) ) THEN
      GOTO 1924
      ENDIF

C WRITE(*,*) AN(TI) , AN(TJ)

C     LIST の配列があふれる時 STOP

      IF ( ( LIST_VDW(0,TI) .EQ. MAX_LIST_VN ) .OR.
     #     ( LIST_VDW(0,TJ) .EQ. MAX_LIST_VN ) ) THEN
         WRITE(*,*) 'LIST_VN is overflow: MAX_LIST_VN.'
         STOP
      ENDIF

      IDX = IDX + 1
      KYORI(IDX) = THP_R

      LIST_VDW(0,TI) = LIST_VDW(0,TI) + 1
      LIST_VDW( LIST_VDW(0,TI) ,TI ) = IDX
      LIST_VDW( LIST_VDW(0,TI) + MAX_LIST_VN ,TI ) = TJ

      LIST_VDW(0,TJ) = LIST_VDW(0,TJ) + 1
      LIST_VDW( LIST_VDW(0,TJ) ,TJ ) = IDX
      LIST_VDW( LIST_VDW(0,TJ) + MAX_LIST_VN ,TJ ) = TI

C
C     カットオフ関数を計算
C

      THP_CUTOFF = 0.0D0
      IF ( THP_R .GE. ( PRH_VCR - PRH_VCD) ) THEN
         THP_CUTOFF = 0.5D0 *
     #   ( 1.0D0 + SIN( PI * ( THP_R - PRH_VCR ) / (2.0D0 * PRH_VCD ) ))
      ENDIF
      IF ( THP_R .GE. ( PRH_VCR + PRH_VCD) ) THEN
         THP_CUTOFF = 1.0D0
      ENDIF
      IF ( THP_R .GE. ( PRH_VCR2 - PRH_VCD2) ) THEN
         THP_CUTOFF = 0.5D0 *
     #  ( 1.0D0 - SIN( PI * (THP_R - PRH_VCR2 )/(2.0D0 * PRH_VCD2 ) ))
      ENDIF
      IF ( THP_R .GE. ( PRH_VCR2 + PRH_VCD2) ) THEN
         THP_CUTOFF = 0.0D0
      ENDIF

      CUTOFF(IDX) = THP_CUTOFF

 1924  CONTINUE
 1925  CONTINUE

 1931  CONTINUE
 1932  CONTINUE
 1933  CONTINUE

 1941  CONTINUE
 1942  CONTINUE
 1943  CONTINUE


C      DO 1952 I = 1 , N
C         write(*,*) "--",I
C         DO 1951 J = 1 , LIST_VDW(0,I)
C            WRITE(*,*) LIST_VDW(J + MAX_LIST_VN , I)
C 1951     CONTINUE
C 1952  CONTINUE

 1969  RETURN
 1999  STOP
      END

C
C 原子 I に関した Van Der Waals ポテンシャル
C
 2001 REAL*8 FUNCTION VDW_I(I,ZAHYO,LIST_VDW,KYORI,CUTOFF)
      INCLUDE 'PARAMETER'
      INTEGER I

      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      REAL*8  R,U
      REAL*8 VDW_FUNC

      INTEGER L,IDXJ
```

```
      U = 0.0D0

      DO 2010 L = 1 , LIST_VDW(0,I)
         IDXJ = LIST_VDW(L,I)
         R = KYORI(IDXJ)
         U = U + CUTOFF(IDXJ) * VDW_FUNC(R)

 2010 CONTINUE

      VDW_I = U
C      write(*,*) "VDW" , U , R

      RETURN

 2049 STOP
      END

 2050 REAL*8 FUNCTION VDW_FUNC(R)
      INCLUDE 'PARAHETER'
      REAL*8 R,SR
      REAL*8 THP_CUTOFF
      REAL*8 SIN


      SR = 3.407D0 / R


      THP_CUTOFF = 0.0D0
      IF ( R .GE. ( PRH_VCR - PRH_VCD) ) THEN

      THP_CUTOFF = 0.5D0 *
     #  ( 1.0D0 + SIN( PI * ( R - PRH_VCR ) / (2.0D0 * PRH_VCD ) ))
      ENDIF
      IF ( R .GE. ( PRH_VCR + PRH_VCD) ) THEN
         THP_CUTOFF = 1.0D0
      ENDIF
      IF ( R .GE. ( PRH_VCR2 - PRH_VCD2) ) THEN
         THP_CUTOFF = 0.5D0 *
     #   ( 1.0D0 - SIN( PI * ( R - PRH_VCR2 ) / (2.0D0 * PRH_VCD2 ) ))
      ENDIF
      IF ( R .GE. ( PRH_VCR2 + PRH_VCD2) ) THEN
         THP_CUTOFF = 0.0D0
      ENDIF


      VDW_FUNC =
     #     4.0D0 * 0.002964D0 *
     #     ( SR**12.0D0 - SR**6.0D0 )
     #     * THP_CUTOFF

      RETURN
 2099 STOP
      END


 2101 SUBROUTINE CALC_DIFF1_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF1)
      INCLUDE 'PARAHETER'
      INTEGER N,N3
      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      REAL*8  DIFF1(MAX_ATOM3)

      INTEGER I,ID3
      REAL*8  ZAHYO_ORG
      REAL*8 VDW_I

      REAL*8 THP

      N3 = N * 3

      DO 2110 I = 1 , N3

         ID3 = ( I + 2 ) / 3
         ZAHYO_ORG = ZAHYO(I)

         ZAHYO(I) = ZAHYO_ORG + EPS1
         CALL RECALC_VDW(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)
         THP = VDW_I(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)

         ZAHYO(I) = ZAHYO_ORG - EPS1
         CALL RECALC_VDW(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)
         THP = THP -
     #        VDW_I(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)

         ZAHYO(I) = ZAHYO_ORG
         CALL RECALC_VDW(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)

         THP = THP / ( 2.0 * EPS1 )
```

```
C         WRITE(*,*) , I , THP
         DIFF1(I) = DIFF1(I) - THP
C         DIFF1(I) = - THP



 2110  CONTINUE

      RETURN
 2199 STOP
      END
 2201 SUBROUTINE RECALC_VDW(I,ZAHYO,LIST_VDW,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I,I3
      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER J,J3
      INTEGER IDXJ
      INTEGER H

      REAL*8 THP_R
      REAL*8 THP_CUTOFF

C      REAL*8 SIN
      REAL*8 SQRT

      I3 = I*3-2

      DO 2210 H = 1 , LIST_VDW(0,I)
         IDXJ = LIST_VDW(H,I)
         J = LIST_VDW(H+MAX_LIST_VN,I)

         J3 = J*3-2
          THP_R =
     #    SQRT    ( ( ZAHYO(J3) - ZAHYO(I3) )**2 +
     #        ( ZAHYO(J3+1) - ZAHYO(I3+1) )**2 +
     #        ( ZAHYO(J3+2) - ZAHYO(I3+2) )**2  )


         KYORI(IDXJ) = THP_R

         THP_CUTOFF = 1.0D0
C         IF ( THP_R .GE. ( PRM_CR - PRM_CD) ) THEN
C          THP_CUTOFF = 0.5D0 *
C     #   ( 1.0D0 - SIN( PI * ( THP_R - PRM_CR ) / (2.0D0 * PRM_CD ) ))
C          ENDIF
C         IF ( THP_R .GE. ( PRM_CR + PRM_CD) ) THEN
C            THP_CUTOFF = 0.0D0
C         ENDIF

         CUTOFF(IDXJ) = THP_CUTOFF

 2210 CONTINUE

      RETURN
 2299 STOP
      END
 2301  REAL*8 FUNCTION VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I
      INTEGER N
      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)
      REAL*8  VDW_I

      VDW = 0.0D0

      DO 2310 I = 1 , N
         VDW = VDW +
     #       VDW_I(I,ZAHYO,LIST_VDW,KYORI,CUTOFF)

 2310  CONTINUE

      VDW = VDW / 2.0D0

      RETURN
 2399  STOP
      END

 2401 SUBROUTINE CALC_DIFF2_VDW
     #(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
      INCLUDE 'PARAHETER'

      INTEGER N,N3
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)

      REAL*8  ZAHYO(MAX_ATOM3)
```

```
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)

      INTEGER I,J,K
      INTEGER L

      N3 = N * 3

C      DO 2410 I = 1 , MAX_ATOM3
C         DIFF2(0,I) = 0
C 2410 CONTINUE

      DO 2450 I = 1 , N

C     隣合う原子の座標系
         DO 2445 L = 1 , LIST_VDW(0,I)
            J = LIST_VDW(L+MAX_LIST_VN,I)

            CALL CALC_DIFF2_VDW_1
     #           (I,J,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
 2445    CONTINUE


 2450 CONTINUE

      RETURN
 2499 STOP
      END

 2501 SUBROUTINE CALC_DIFF2_VDW_1
     #     (I,J,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

      INCLUDE 'PARAHETER'
      INTEGER I,J
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)

      REAL*8  ZAHYO(MAX_ATOM3)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)

      REAL*8  VDW_FUNC

      INTEGER I3 , J3
      INTEGER II,JJ
      INTEGER I3II,J3JJ
      REAL*8 X,Y

      REAL*8 THP
      REAL*8 DATA

      REAL*8 R,DIST

      REAL*8 GI(3)
      REAL*8 GJ(3)

      I3 = I*3 - 3
      J3 = J*3 - 3

      GI(1) = ZAHYO(I3 + 1)
      GI(2) = ZAHYO(I3 + 2)
      GI(3) = ZAHYO(I3 + 3)

      GJ(1) = ZAHYO(J3 + 1)
      GJ(2) = ZAHYO(J3 + 2)
      GJ(3) = ZAHYO(J3 + 3)


      DIST = CALC_R(GI,GJ)

      DO 2530 II = 1 , 3
         DO 2520 JJ = 1 , 3


               X = GI(II)
               Y = GJ(JJ)

               GI(II) = X + EPS2
               GJ(JJ) = Y + EPS2
               R = CALC_R(GI,GJ)

               THP = VDW_FUNC(R)

               GI(II) = X - EPS2
               GJ(JJ) = Y - EPS2
               R = CALC_R(GI,GJ)
```

```
              THP = THP +  VDW_FUNC(R)


              GI(II) = X + EPS2
              GJ(JJ) = Y - EPS2
              R = CALC_R(GI,GJ)

              THP = THP -  VDW_FUNC(R)



              GI(II) = X - EPS2
              GJ(JJ) = Y + EPS2
              R = CALC_R(GI,GJ)

              THP = THP - VDW_FUNC(R)

              DATA = THP / ( 8.0D0 * EPS2 * EPS2 )


              IF ( DIST .LT. (PRH_CR+PRH_CD) ) THEN
             CALL ADD2_DIFF2( DATA , I3+II , J3+JJ , DIFF2 ,DIFF2_DATA)
              ELSE
             CALL ADD_DIFF2( DATA , I3+II , J3+JJ , DIFF2 ,DIFF2_DATA)
              ENDIF
              GI(II) = X
              GJ(JJ) = Y


 2520    CONTINUE
 2530 CONTINUE

      RETURN
 2599 STOP
      END


 2601 REAL*8 FUNCTION CALC_R(GI,GJ)

      REAL*8 GI(3)
      REAL*8 GJ(3)
      REAL*8 SQRT


      CALC_R = SQRT(
     #     (GI(1) - GJ(1))**2+
     #     (GI(2) - GJ(2))**2+
     #     (GI(3) - GJ(3))**2 )




      RETURN
 2699 STOP
      END

2701  SUBROUTINE HAKE_NEWKYORI(N,ZAHYO,LIST,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER N,I
      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

DO 2710 I = 1 , N
CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
 2710  CONTINUE

      RETURN
 2799  STOP
      END

2801  SUBROUTINE HAKE_NEWKYORIV(N,ZAHYO,LIST_VDW,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER N,I
      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST_VDW(0:HAX_LIST_VN2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

DO 2810 I = 1 , N
CALL RECALC_VDW(I,ZAHYO,LIST_VDW,KYORI,CUTOFF)
 2810  CONTINUE

      RETURN
 2899  STOP
      END
```

B.2 ２層構造の結合エネルギーを計算するプログラム

本研究において作成した、2 層構造ナノチューブの互いの軸方向のずれや回転のずれに関して層間の potential を計算するプログラムを示す。

使い方は

```
ir5 inner.xyz outer.xyz >! kekka.xy
```

であるが 内側のユニットセルの格子定数 /2 を 0.01Å 単位で記述しておく必要がある。格子定数が 18.56Å なら、 9280 である。

行番号 35 付近の

```
      DO 43 SL =  -9280 , 9280 , 40
          DO 42 DEG = -180 , 180 , 1
```

を変更する。

```
c
c     Tersoff potential for Carbon system
c     (1998/Jun/8)   By  R.Hatsuo .
c

      PROGRAH innerrotate4
      INCLUDE 'PARAHETER'

C     原子座標の FILE 名変数
      CHARACTER*50 FILENAHE
      CHARACTER*50 FILENAHE2

C     原子数の保持変数
      INTEGER N
      INTEGER N1
      INTEGER N2


C     loop 用変数
      INTEGER I,J,K

C     座標用配列
      REAL*8  ZAHYO(HAX_ATOH3)
      REAL*8  ZAHYO1(HAX_ATOH3)
      REAL*8  ZAHYO2(HAX_ATOH3)

      CHARACTER*8  AN(HAX_ATOH)
      CHARACTER*8  AN1(HAX_ATOH)
      CHARACTER*8  AN2(HAX_ATOH)


      REAL*8 ZL,ZH

C     近接リスト配列
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)

C     第二２近接リスト配列
      INTEGER LIST2(0:HAX_LIST2_N,HAX_ATOH)

C     近接リスト配列
      INTEGER LIST_VDW(0:HAX_LIST_VN2,HAX_ATOH)

C     近接データ配列 IDX でリストから参照される
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      REAL*8  DIFF1(HAX_ATOH3)
      REAL*8  VELO(HAX_ATOH3)

C     TERSOFF 関数の型
      REAL*8 TERSOFF

C     系のエネルギー
      REAL*8 ENERGY , ENERGY_VDW

      INTEGER IDX
      INTEGER SL,DEG
```

```
C     FILE名を取得する
      CALL GETFILENAHE(FILENAHE,FILENAHE2)

      open(unit=20,file='md.log',ACCESS='APPEND',status='old')
      write(20,25) FILENAHE,FILENAHE2

 25   format(a50)

C     XYZ座標を読み込む
      CALL READ_ZAHYO(FILENAHE, N1, ZAHYO1,AN1)
      CALL READ_ZAHYO(FILENAHE2, N2, ZAHYO2,AN2)

      ZL = ZAHYO1(3)
      ZH = ZAHYO1(3)

      DO 26 I = 3 , N1*3 , 3
         IF ( ZAHYO1(I) .GT. ZH) THEN
            ZH = ZAHYO1(I)
         ENDIF
         IF ( ZAHYO1(I) .LT. ZL) THEN
            ZL = ZAHYO1(I)
         ENDIF
 26   CONTINUE

      DO 27 I = 3 , N1*3 , 3
         ZAHYO1(I) =  ZAHYO1(I) - ( ZH + ZL  ) / 2
 27   CONTINUE


      ZL = ZAHYO2(3)
      ZH = ZAHYO2(3)

      DO 28 I = 3 , N2*3 , 3
         IF ( ZAHYO2(I) .GT. ZH) THEN
            ZH = ZAHYO2(I)
         ENDIF
         IF ( ZAHYO2(I) .LT. ZL) THEN
            ZL = ZAHYO2(I)
         ENDIF
 28   CONTINUE

      DO 29 I = 3 , N2*3 , 3
         ZAHYO2(I) =  ZAHYO2(I) - ( ZH + ZL ) /2
 29   CONTINUE


      DO 30 I = 1 , N1*3
         ZAHYO(I) = ZAHYO1(I)
 30   CONTINUE

      DO 31 I = 1 , N1
         AN(I) = AN1(I)
 31   CONTINUE

      DO 32 I = 1 , N2*3
         ZAHYO(N1*3+I) = ZAHYO2(I)

 32   CONTINUE

      DO 33 I = 1 , N2
         AN(N1+I) = AN2(I)
 33   CONTINUE

      N = N1 + N2

      DO 34 I = 1 , N*3
         ZAHYO2(I) = ZAHYO(I)
 34   CONTINUE


      DO 35 J = 1 , MAX_ATOM3
         VELO(J) = 0.0D0
 35   CONTINUE

      IDX = 0

         DO 43 SL =  -9280 , 9280 , 40
            DO 42 DEG = -180 , 180 , 1

                 DO 36 I = 1 , N*3
               ZAHYO(I) = ZAHYO2(I)
 36         CONTINUE

            CALL INNERROTATE(N,ZAHYO,AN,-DEG)
C            CALL INNERSLIDE(N,ZAHYO,AN,(SL-1000))
             CALL INNERSLIDE(N,ZAHYO,AN,(SL))

       CALL  PRINT_ZAHYO(N,ZAHYO,0,ENERGY,DIFF1,AN)
       STOP
       DO 41 I = 1 , 1
```

```
      IDX = 0

C      write(*,*) 'make_list'
C      CALL HAKE_LIST(N,ZAHYO,LIST,KYORI,CUTOFF,IDX)

C     第二近接リストを生成

C       write(*,*) 'make_list2'
C      CALL HAKE_LIST2(N,LIST,LIST2)

C      write(*,*) IDX

C       write(*,*) 'make_list_vdw'
      CALL HAKE_LIST_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,IDX,LIST,
     #LIST2,AN)

C      write(*,*) IDX

C       write(*,*) 'make_diff'
C       CALL CALC_DIFF1(N,ZAHYO,LIST,KYORI,CUTOFF,DIFF1)

C        write(*,*) 'make_diff_vdw'
C       CALL CALC_DIFF1_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF1)

C      DO 39 J = 1 , N*3
C          VELO(J) = VELO(J) + DIFF1(J) * CONV_VELO
C         ZAHYO(J) = ZAHYO(J) + VELO(J)
C          VELO(J) = VELO(J) * 0.9D0

 39    CONTINUE

C       ENERGY = TERSOFF(N,ZAHYO,LIST,KYORI,CUTOFF)

       ENERGY_VDW = VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF)
       ENERGY =  ENERGY_VDW /N2

C WRITE(*,*) ENERGY
       WRITE(20,40) I , ENERGY
 40   format(i6,f20.7)

 41   CONTINUE
      WRITE(*,44) DEG , 1.0D0 * SL / 1000.0D0  , ENERGY
C       CALL  PRINT_ZAHYO(N,ZAHYO,0,ENERGY,DIFF1,AN)

C      WRITE(*,*) SL ,  ENERGY


 42   CONTINUE
 43   CONTINUE
 44   format(i4,f14.8,f20.10)
C 44   format(f14.8,f20.10)
      close(20)
      STOP
      END

c
c This program is for md to run on Dec Fortran
c

C      function IARGC()
C      IARGC=1
C      return
C      end
C      subroutine GETARG(i,c)
C      character*80 c
C      open(unit=53,file='jobname.dat',status='old')
C      read(53,45) c
C 45   format(a30)
C      close(53)
C      return
C      end

C      subroutine FDATE(IDATE)
C      CHARACTER IDATE*24
C      call DATE(IDATE)
C      return
C      end

CC
C     座標データの FILE 名を取得する サブルーチン
CC

 101  SUBROUTINE GETFILENAHE(FILENAHE,FILENAHE2)
      CHARACTER*50 FILENAHE,FILENAHE2
      INTEGER I
      I = IARGC()
      IF (I .NE. 2) then
         WRITE(*,*) 'Usage: ir hoge.xyz hoge2.xyz'
         GOTO 110
      ENDIF
```

```
      CALL GETARG(1,FILENAHE)
      CALL GETARG(2,FILENAHE2)


      RETURN
110   END


121   SUBROUTINE GETARGU(LINE,I,ARGU)
      CHARACTER*80 LINE
      CHARACTER*80 DUH
      CHARACTER*80 ARGU
      INTEGER I
      INTEGER J
      INTEGER K
      INTEGER lenline

      DUH = LINE


      do 128 K = 1 , I
         lenline = len(DUH)
         DO 122 J = 1 , lenline

            IF ( DUH(J:J) .EQ. ' ') THEN
               GOTO 122
            ENDIF
            IF ( DUH(J:J) .EQ. char(9) ) THEN
               GOTO 122
            ENDIF
            IF ( DUH(J:J) .EQ. char(0) ) THEN
               GOTO 122
            ENDIF
            GOTO 123
122      CONTINUE

123      DUH = DUH(J:lenline)

         lenline = len(DUH)
         DO 124 J = 2 , lenline
            IF ( DUH(J:J) .EQ. ' ') THEN
               GOTO 125
            ENDIF
            IF ( DUH(J:J) .EQ. char(9) ) THEN
               GOTO 125
            ENDIF
            IF ( DUH(J:J) .EQ. char(0) ) THEN
               GOTO 125
            ENDIF
124      CONTINUE
125      ARGU = DUH(1:J-1)

         DUH=DUH(J:lenline)

128   continue

      RETURN

129   END

130   REAL*8 FUNCTION ATOR(STR)
      CHARACTER*80 STR
      INTEGER I
      INTEGER J
      INTEGER S
      INTEGER D
      INTEGER DCU
      INTEGER*4 R
      INTEGER*4 E

      I = 1
      S = 1

      R = 0
      EC = 0
      E = 0

      IF ( STR(1:1) .EQ. '+') THEN
         S = 1
         I = I + 1
      ENDIF
      IF ( STR(1:1) .EQ. '-') THEN
         S = -1
         I = I + 1
      ENDIF

      DO 140 J = I , 80
         IF ( STR(J:J) .EQ. '.' ) THEN
            EC = 1
            GOTO 140
         ENDIF
```

```
         D = ICHAR(STR(J:J)) - ICHAR('0')

         IF ( (D .GT. 9) .OR. (D .LT. 0)) THEN
         GOTO 145
      ENDIF
      R = R * 10 + D
      E = E + EC

      IF ( R .GT. 9999999 ) THEN
         GOTO 145
      ENDIF

 140  CONTINUE

 145  ATOR = 1.0D0  * S * R / (10.0D0**E)

      RETURN

 149  END

CC
C     座標データ取り込み サブルーチン
CC

 201  SUBROUTINE READ_ZAHYO(FILENAHE, N, ZAHYO,AN)
      INCLUDE 'PARAHETER'
      CHARACTER*50 FILENAHE

C     原子数を保持する変数
      INTEGER N
      INTEGER N3

C     FILE IO チェック用変数
      INTEGER IOCHECK

C     loop variable
      INTEGER I

C     炭素原子の座標用配列
      REAL*8 ZAHYO(HAX_ATOH3)

C     元素名用変数
      CHARACTER*8 AN(HAX_ATOH)

      CHARACTER*80 LINE,ARGU

C     FILE OEPN
      OPEN(60,FILE=FILENAHE,STATUS='OLD',IOSTAT=IOCHECK)

C     FILE OPEN のエラーチェック
      IF ( IOCHECK ) THEN
         WRITE(*,*) 'File open error.'
         GOTO 299
      ENDIF

C     原子数の制限チェック
      READ(60,*) N

      IF ( HAX_ATOH .LT. N ) THEN
         WRITE(*,*) "Too many atoms."
         GOTO 299
      ENDIF

      N3 = N * 3

      READ(60,209) LINE

 208  format(f14.8)
 209  format(a80)

C     座標読み込み 単位はÅ
      DO 210 I = 1 , N3 , 3
         READ(60,209) LINE
         CALL GETARGU(LINE,1,ARGU)
         AN(I/3+1)=ARGU
         CALL GETARGU(LINE,2,ARGU)
         ZAHYO(I) = ATOR(ARGU)
         CALL GETARGU(LINE,3,ARGU)
         ZAHYO(I+1) = ATOR(ARGU)
         CALL GETARGU(LINE,4,ARGU)
         ZAHYO(I+2) = ATOR(ARGU)

C         READ(60,*) AN(I/3+1) , ZAHYO(I) ,ZAHYO(I+1) ,ZAHYO(I+2)
 210  CONTINUE

      CLOSE(60,STATUS='KEEP')
      RETURN

 299  CLOSE(60,STATUS='KEEP')
      END

 301  SUBROUTINE PRINT_ZAHYO(N,ZAHYO,T,ENERGY,DIFF1,AN)
```

```
      INCLUDE 'PARAHETER'
      INTEGER N,N3
      INTEGER T
      REAL*8 ZAHYO(HAX_ATOH3)
      CHARACTER*8 AN(HAX_ATOH)
      REAL*8 DIFF1(HAX_ATOH3)
      REAL*8 ENERGY
      INTEGER I

      WRITE(*,*) N
      WRITE(*,*) 'TIHE=',T*TIHE_DIV*1.0D15,'fs  ENERGY =',ENERGY

      N3 = N * 3

      DO 311 I = 1 , N3 , 3
         WRITE(*,305) AN(I/3+1) , ZAHYO(I) , ZAHYO(I+1) , ZAHYO(I+2)
     #    , DIFF1(I) *10.0 , DIFF1(I+1) * 10.0 , DIFF1(I+2) * 10.0

 305     FORHAT(a5,f12.7,f12.7,f12.7,f12.7,f12.7,f12.7)
 311  CONTINUE

 398  RETURN
 399  END
CC
C     最近接原子の LIST を作成するサブルーチン
CC

C
C     LIST(0,N) : 原子番号 N の近接原子の数
C     LIST(I,N) : 原子番号 N の I 番目の近接原子インデックス
C     LIST(I+HAX_LIST_N ,N) : 原子番号 N の I 番目の近接原子番号

 401  SUBROUTINE HAKE_LIST(N,ZAHYO,LIST,KYORI,CUTOFF,IDX)
      INCLUDE 'PARAHETER'
      INTEGER N,N3

      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)
      INTEGER IDX

      REAL*8  HAX_X,HAX_Y,HAX_Z
      REAL*8  HIN_X,HIN_Y,HIN_Z

      INTEGER I,K,J
      INTEGER E,F,G
      INTEGER L,H

      INTEGER I3

      INTEGER HESH_X,HESH_Y,HESH_Z
      INTEGER TX,TY,TZ
      INTEGER TI,TJ,TI3,TJ3

      INTEGER HESH(0:HAX_HESH_ATOH,
     #       0:HAX_HESH_X+1,0:HAX_HESH_Y+1,0:HAX_HESH_Z+1)

      REAL*8 THP_R,THP_CUTOFF

      REAL*8 SIN

      N3 = N * 3

C     リスト配列の初期化
      DO 405 I = 1 , HAX_ATOH
         LIST(0,I) = 0
 405  CONTINUE

C     HESH 配列の初期化
      DO 406 I = 0 , HAX_HESH_X+1
         DO 406 J = 0 , HAX_HESH_Y+1
            DO 406 K = 0 , HAX_HESH_Z+1
               HESH(0,I,J,K) = 0
 406  CONTINUE

C     系の座標の最大値 - 最小値を検索する
      HAX_X = ZAHYO(1)
      HAX_Y = ZAHYO(2)
      HAX_Z = ZAHYO(3)
      HIN_X = ZAHYO(1)
      HIN_Y = ZAHYO(2)
      HIN_Z = ZAHYO(3)

      DO 410 I = 4 , N3 , 3

         IF ( HAX_X .LT. ZAHYO(I) ) THEN
            HAX_X = ZAHYO(I)
         ENDIF
         IF ( HAX_Y .LT. ZAHYO(I+1) ) THEN
            HAX_Y = ZAHYO(I+1)
         ENDIF
```

```
         IF ( HAX_Z .LT. ZAHYO(I+2) ) THEN
            HAX_Z = ZAHYO(I+2)
         ENDIF
         IF ( HIN_X .GT. ZAHYO(I) ) THEN
            HIN_X = ZAHYO(I)
         ENDIF
         IF ( HIN_Y .GT. ZAHYO(I+1) ) THEN
            HIN_Y = ZAHYO(I+1)
         ENDIF
         IF ( HIN_Z .GT. ZAHYO(I+2) ) THEN
            HIN_Z = ZAHYO(I+2)
         ENDIF
 410  CONTINUE

C     一度、大まかに原子をグルーピングする。
C     最大値最小値から、メッシュの個数を決める

      HESH_X = 1 + ( ( HAX_X - HIN_X ) / HESH_D )
      HESH_Y = 1 + ( ( HAX_Y - HIN_Y ) / HESH_D )
      HESH_Z = 1 + ( ( HAX_Z - HIN_Z ) / HESH_D )

C     HESH 用の配列 に収まらない場合 STOP
      IF ( HESH_X .LE. HAX_HESH_X ) THEN
         GOTO 415
      ENDIF
      IF ( HESH_Y .LE. HAX_HESH_Y ) THEN
         GOTO 415
      ENDIF
      IF ( HESH_Z .LE. HAX_HESH_Z ) THEN
         GOTO 415
      ENDIF

      WRITE(*,*) 'HESH is overflow: HAX_HESH.',  HESH_X,HESH_Y,HESH_Z
      GOTO 499

C     全ての原子を HESH に グルーピングする。

 415  DO 420 I = 1 , N
         I3 = I * 3 - 2
         TX = 1 + ( ( ZAHYO(I3) - HIN_X ) / HESH_D )
         TY = 1 + ( ( ZAHYO(I3+1) - HIN_Y ) / HESH_D )
         TZ = 1 + ( ( ZAHYO(I3+2) - HIN_Z ) / HESH_D )

C     HESH 配列があるれる場合 STOP
         IF ( HESH(0,TX,TY,TZ) .EQ. HAX_HESH_ATOH ) THEN
            WRITE(*,*) 'HESH_ATOH is overflow: HAX_HESH_ATOH.'
            GOTO 499
         ENDIF

         HESH(0,TX,TY,TZ) =  HESH(0,TX,TY,TZ) + 1
         HESH( HESH(0,TX,TY,TZ) , TX,TY,TZ) = I

 420  CONTINUE


C     HESH のブロックの 近接内 (27 ブロック)
C     で 距離を計算し 近接LIST を作成する
C     ついでに カットオフ関数の値も計算する

      DO 443 I = 1 , HESH_X
         DO 442 J = 1 , HESH_Y
            DO 441 K = 1 , HESH_Z

               IF ( HESH(0,I,J,K) .EQ. 0 ) THEN
                  GOTO 441
               ENDIF

               DO 433 E = I - 1  , I + 1
                  DO 432 F = J - 1 ,  J + 1
                     DO 431 G = K - 1  , K + 1

                        IF ( HESH(0,E,F,G) .EQ. 0 ) THEN
                           GOTO 431
                        ENDIF

                        DO 425 L = 1 , HESH(0,I,J,K)
                           DO 424 H = 1 , HESH(0,E,F,G)

                              TI = HESH(L,I,J,K)
                              TJ = HESH(H,E,F,G)

                              IF ( TI .LE. TJ ) THEN
                                 GOTO 424
                              ENDIF

                              TI3 = TI*3-2
                              TJ3 = TJ*3-2
C
C     原子間距離の計算
C
      THP_R =
     #     ( ( ZAHYO(TJ3  ) - ZAHYO(TI3  ) )**2 +
```

```
     #       ( ZAHYO(TJ3+1) - ZAHYO(TI3+1) )**2 +
     #       ( ZAHYO(TJ3+2) - ZAHYO(TI3+2) )**2  )**0.5d0

C
C     近接でない場合場合 処理を飛ばす
C
      IF ( THP_R .GT. (PRH_CR + PRH_CD) ) THEN
         GOTO 424
      ENDIF

C     LIST の配列があふれる時 STOP

      IF ( ( LIST(0,TI) .EQ. HAX_LIST_N ) .OR.
     #     ( LIST(0,TJ) .EQ. HAX_LIST_N ) ) THEN
         WRITE(*,*) 'LIST_N is overflow: HAX_LIST_N.'
         GOTO 499
      ENDIF

      IDX = IDX + 1
      KYORI(IDX) = THP_R

      LIST(0,TI) = LIST(0,TI) + 1
      LIST( LIST(0,TI) ,TI ) = IDX
      LIST( LIST(0,TI) + HAX_LIST_N ,TI ) = TJ

      LIST(0,TJ) = LIST(0,TJ) + 1
      LIST( LIST(0,TJ) ,TJ ) = IDX
      LIST( LIST(0,TJ) + HAX_LIST_N ,TJ ) = TI

C
C     カットオフ関数を計算
C

      THP_CUTOFF = 1.0D0
      IF ( THP_R .GE. ( PRH_CR - PRH_CD) ) THEN
         THP_CUTOFF = 0.5D0 *
     #   ( 1.0D0 - SIN( PI * ( THP_R - PRH_CR ) / (2.0D0 * PRH_CD ) ))
      ENDIF
      IF ( THP_R .GE. ( PRH_CR + PRH_CD) ) THEN
         THP_CUTOFF = 0.0D0
      ENDIF

      CUTOFF(IDX) = THP_CUTOFF

 424  CONTINUE
 425  CONTINUE


 431  CONTINUE
 432  CONTINUE
 433  CONTINUE

 441  CONTINUE
 442  CONTINUE
 443  CONTINUE

C      DO 452 I = 1 , N
C         write(*,*) "--",I
C         DO 451 J = 1 , LIST(0,I)
C            WRITE(*,*) LIST(J + HAX_LIST_N , I)
C 451     CONTINUE
C 452  CONTINUE

 469  RETURN
 499  END
C
C     i 原子に関する Tersoff ポテンシャル
C

 501  REAL*8 FUNCTION TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I

      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      INTEGER J,K,L,H
      INTEGER I3,J3,K3
      INTEGER IDXJ,IDXK

      REAL*8 G,GCOS
      REAL*8 ZETA
      REAL*8 I_X,I_Y,I_Z
      REAL*8 J_X,J_Y,J_Z
      REAL*8 K_X,K_Y,K_Z
      REAL*8 Bij

      REAL*8 EXP

      REAL*8 THP
```

```
      I3 = I*3 - 2
      I_X = ZAHYO(I3)
      I_Y = ZAHYO(I3+1)
      I_Z = ZAHYO(I3+2)

      THP = 0.0D0

      DO 520 L = 1 , LIST(0,I)
         IDXJ = LIST(L,I)
         J = LIST(L+HAX_LIST_N,I)
         J3 = J*3 - 2
         J_X = ZAHYO(J3)
         J_Y = ZAHYO(J3+1)
         J_Z = ZAHYO(J3+2)

         ZETA = 0.0D0

         DO 510 H = 1 , LIST(0,I)
            IF ( L .EQ. H ) THEN
               GOTO 510
            ENDIF

            IDXK = LIST(H,I)
            K = LIST(H+HAX_LIST_N,I)
            K3 = K*3 - 2
            K_X = ZAHYO(K3)
            K_Y = ZAHYO(K3+1)
            K_Z = ZAHYO(K3+2)

            GCOS =
     #           (
     #           (J_X - I_X ) * (K_X - I_X )+
     #           (J_Y - I_Y ) * (K_Y - I_Y )+
     #           (J_Z - I_Z ) * (K_Z - I_Z )
     #           ) / KYORI(IDXJ) / KYORI(IDXK)

            IF ( KYORI(IDXK) .EQ. 0.0D0 ) THEN
               WRITE(*,*) KYORI(IDXK) ,I,J,K
            ENDIF


            G = ( PRH_C * PRH_C ) / ( PRH_D * PRH_D ) -
     #           ( PRH_C * PRH_C ) /
     #           ( PRH_D * PRH_D + ( PRH_H - GCOS ) * (PRH_H - GCOS ))

            G = PRH_A * ( G + 1.0D0 )
            ZETA = ZETA + CUTOFF(IDXK) * G

510      CONTINUE

         Bij = ( 1.0 + ( ZETA )**( PRH_ETA) )**( -PRH_DELTA )

         THP = THP +
     #   CUTOFF(IDXJ) *
     #   (
     #    PRH_EA * EXP( -PRH_LUHBDA1 * KYORI(IDXJ) ) +
     #    (PRH_EB) * Bij * EXP( -PRH_LUHBDA2 * KYORI(IDXJ))
     #   )

520   CONTINUE

      TERSOFF_I = THP

      RETURN
599   END

601   REAL*8 FUNCTION TERSOFF(N,ZAHYO,LIST,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I
      INTEGER N
      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)
      REAL*8  TERSOFF_I

      TERSOFF = 0.0D0

      DO 610 I = 1 , N
         TERSOFF = TERSOFF +
     #        TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)

610   CONTINUE

      TERSOFF = TERSOFF / 2.0D0

      RETURN
699   END

701   SUBROUTINE CALC_DIFF1(N,ZAHYO,LIST,KYORI,CUTOFF,DIFF1)
      INCLUDE 'PARAHETER'
```

```
      INTEGER N,N3
      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      REAL*8  DIFF1(HAX_ATOH3)

      INTEGER I,ID3
      REAL*8  ZAHYO_ORG
      REAL*8 TERSOFF_LI

      REAL*8 THP

      N3 = N * 3

      DO 710 I = 1 , N3
         ID3 = ( I + 2 ) / 3
         ZAHYO_ORG = ZAHYO(I)

         ZAHYO(I) = ZAHYO_ORG + EPS1
         CALL RECALC(ID3,ZAHYO,LIST,KYORI,CUTOFF)
         THP = TERSOFF_LI(ID3,ZAHYO,LIST,KYORI,CUTOFF)

         ZAHYO(I) = ZAHYO_ORG - EPS1
         CALL RECALC(ID3,ZAHYO,LIST,KYORI,CUTOFF)
         THP = THP -
     #        TERSOFF_LI(ID3,ZAHYO,LIST,KYORI,CUTOFF)

         ZAHYO(I) = ZAHYO_ORG
         CALL RECALC(ID3,ZAHYO,LIST,KYORI,CUTOFF)

         THP = THP / ( 2.0 * EPS1 )

         DIFF1(I) = -THP

710   CONTINUE

      RETURN
799   END

801   SUBROUTINE RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I,I3
      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      INTEGER J,J3
      INTEGER IDXJ
      INTEGER H

      REAL*8 THP_R
      REAL*8 THP_CUTOFF

      REAL*8 SIN
      REAL*8 SQRT

      I3 = I*3-2

      DO 810 H = 1 , LIST(0,I)
         IDXJ = LIST(H,I)
         J = LIST(H+HAX_LIST_N,I)
         IDXJ = LIST(H,I)

         J3 = J*3-2
          THP_R =
     #    SQRT    ( ( ZAHYO(J3) - ZAHYO(I3) )**2 +
     #         ( ZAHYO(J3+1) - ZAHYO(I3+1) )**2 +
     #         ( ZAHYO(J3+2) - ZAHYO(I3+2) )**2  )


         KYORI(IDXJ) = THP_R

         THP_CUTOFF = 1.0D0
         IF ( THP_R .GE. ( PRH_CR - PRH_CD) ) THEN
         THP_CUTOFF = 0.5D0 *
     #   ( 1.0D0 - SIN( PI * ( THP_R - PRH_CR ) / (2.0D0 * PRH_CD ) ))
         ENDIF
         IF ( THP_R .GE. ( PRH_CR + PRH_CD) ) THEN
            THP_CUTOFF = 0.0D0
         ENDIF

         CUTOFF(IDXJ) = THP_CUTOFF

810   CONTINUE

      RETURN
899   END

901   REAL*8 FUNCTION TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)
```

```
      INCLUDE 'PARAHETER'
      INTEGER I
      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)
      REAL*8  TERSOFF_I
      INTEGER L

      INTEGER J

      TERSOFF_LI = TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)

      DO 910 L = 1 , LIST(0,I)
         J = LIST(L + HAX_LIST_N,I)
         TERSOFF_LI = TERSOFF_LI +
     #        TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)

 910  CONTINUE

      RETURN

 999  END


 1001 SUBROUTINE HAKE_LIST2(N,LIST,LIST2)
      INCLUDE 'PARAHETER'
      INTEGER N
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      INTEGER LIST2(0:HAX_LIST2_N,HAX_ATOH)

      INTEGER I,J,K,L,H

      DO 1010 I = 1 , N
         LIST2(0,I) = 0

         DO 1005 J = 1 , LIST(0,I)
            L = LIST(J+HAX_LIST_N,I)

            DO 1003 K = 1 , LIST(0,L)
               H = LIST(K+HAX_LIST_N,L)
               IF ( H .EQ. I) THEN
                  GOTO 1003
               ENDIF
               IF ( H .EQ. L) THEN
                  GOTO 1003
               ENDIF

               IF ( LIST2(0,I) .EQ. HAX_LIST2_N ) THEN
                  WRITE(*,*) 'HAX_LIST2_N is overflow'
                  STOP
               ENDIF

               LIST2(0,I) = LIST2(0,I) + 1
               LIST2(LIST2(0,I),I) = H

 1003       CONTINUE
 1005    CONTINUE
 1010 CONTINUE

      RETURN
 1099 END

 1101 SUBROUTINE CALC_DIFF2
     #(N,ZAHYO,LIST,LIST2,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
      INCLUDE 'PARAHETER'

      INTEGER N,N3
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      INTEGER LIST2(0:HAX_LIST2_N,HAX_ATOH)

      REAL*8  ZAHYO(HAX_ATOH3)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)

      INTEGER I,J,K
      INTEGER L

      N3 = N * 3

C      DO 1110 I = 1 , HAX_ATOH3
C         DIFF2(0,I) = 0
C 1110 CONTINUE

      DO 1150 I = 1 , N
C     同一原子の座標系
         CALL CALC_DIFF2_1
     #        (I,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
```

```
C     隣合う原子の座標系
         DO 1145 L = 1 , LIST(0,I)
            J = LIST(L+MAX_LIST_N,I)

            CALL CALC_DIFF2_2
     #           (I,J,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
 1145    CONTINUE


C     第二隣接の座標系

         DO 1146 L = 1 , LIST2(0,I)
            J = LIST2(L,I)
            CALL CALC_DIFF2_3
     #           (I,J,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
 1146       CONTINUE

 1150 CONTINUE

      RETURN
 1199 END

 1201 SUBROUTINE CALC_DIFF2_1
     #     (I,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

      INCLUDE 'PARAMETER'
      INTEGER I
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)

      REAL*8  ZAHYO(MAX_ATOM3)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  TERSOFF_LI

      INTEGER I3 , J3
      INTEGER II,JJ
      INTEGER I3II,J3JJ
      REAL*8 X,Y

      REAL*8 THP
      REAL*8 DATA

      I3 = I*3 - 3
      J3 = I3

      DO 1230 II = 1 , 3
         DO 1220 JJ = 1 , 3
            I3II = I3 + II
            J3JJ = J3 + JJ

            IF ( I3II .NE. J3JJ ) THEN

               X = ZAHYO(I3II)
               Y = ZAHYO(J3JJ)

               ZAHYO(I3II) = X + EPS2
               ZAHYO(J3JJ) = Y + EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X - EPS2
               ZAHYO(J3JJ) = Y - EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP +
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X + EPS2
               ZAHYO(J3JJ) = Y - EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP -
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X - EPS2
               ZAHYO(J3JJ) = Y + EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               THP = THP -
     #              TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

               DATA = THP / ( 8.0D0 * EPS2 * EPS2 )

               ZAHYO(I3II) = X
               ZAHYO(J3JJ) = Y

            ELSE

               X = ZAHYO(I3II)

               ZAHYO(I3II) = X + 2.0D0*EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
```

```
              THP = TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

              ZAHYO(I3II) = X - 2.0D0*EPS2
              CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
              THP = THP +
     #             TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

              ZAHYO(I3II) = X + EPS2
              CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
              THP = THP -
     #             TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

              ZAHYO(I3II) = X - EPS2
              CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
              THP = THP -
     #             TERSOFF_LI(I,ZAHYO,LIST,KYORI,CUTOFF)

              DATA = THP / ( 3.0D0 * EPS2 * EPS2 )

              ZAHYO(I3II) = X

            ENDIF

            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL ADD_DIFF2( DATA , I3II , J3JJ , DIFF2 ,DIFF2_DATA)

 1220    CONTINUE
 1230 CONTINUE

      RETURN
 1299 END

 1301 SUBROUTINE ADD_DIFF2( DATA , I3II , J3JJ , DIFF2 , DIFF2_DATA )
      INCLUDE 'PARAHETER'

      INTEGER I3II , J3JJ
      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DATA

      IF ( DIFF2(0,I3II) .EQ. HAX_DIFF2_LIST ) THEN
         WRITE(*,*) 'HAX_DIFF2_LIST is overflow.'
         GOTO 1349
      ENDIF

      DIFF2(0,I3II) = DIFF2(0,I3II) + 1
      DIFF2(DIFF2(0,I3II),I3II) = J3JJ
      DIFF2_DATA(DIFF2(0,I3II),I3II) = DATA
C      WRITE(*,*) I3II , J3JJ, DATA

      RETURN
 1349 END

 1351 SUBROUTINE ADD2_DIFF2( DATA , I3II , J3JJ , DIFF2 , DIFF2_DATA )
      INCLUDE 'PARAHETER'

      INTEGER I3II , J3JJ
      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DATA
      INTEGER I


      DO 1360 I = 1 , DIFF2(0,I3II)
         IF ( DIFF2(I,I3II) .EQ. J3JJ ) THEN
            DIFF2_DATA(I,I3II) = DIFF2_DATA(I,I3II) + DATA
            GOTO 1380
         ENDIF
 1360 CONTINUE

      WRITE(*,*) 'error ADD2_DIFF2'
      GOTO 1399
C     WRITE(*,*) I3II , J3JJ, DATA



 1380 RETURN
 1399 END




 1401 SUBROUTINE CALC_DIFF2_2
     #     (I,J,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

      INCLUDE 'PARAHETER'
      INTEGER I,J
      INTEGER LIST(0:HAX_LIST_N2,HAX_ATOH)
      INTEGER LIST_IJ(0:HAX_LIST_IJ)

      REAL*8  ZAHYO(HAX_ATOH3)
      REAL*8  KYORI(HAX_DATA_IDX)
```

```
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  TERSOFF_I
      REAL*8  TERSOFF_LIJ

      INTEGER I3 , J3
      INTEGER II,JJ
      INTEGER I3II,J3JJ
      REAL*8 X,Y

      REAL*8 THP
      REAL*8 DATA

      CALL MAKE_LIST_IJ( I,J,LIST_IJ,LIST )

      I3 = I*3 - 3
      J3 = J*3 - 3

      DO 1430 II = 1 , 3
         DO 1420 JJ = 1 , 3
            I3II = I3 + II
            J3JJ = J3 + JJ

               X = ZAHYO(I3II)
               Y = ZAHYO(J3JJ)

               ZAHYO(I3II) = X + EPS2
               ZAHYO(J3JJ) = Y + EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)
               THP = TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
     #              +TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)
     #              +TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X - EPS2
               ZAHYO(J3JJ) = Y - EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

               THP = THP
     #              +TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
     #              +TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)
     #              +TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X + EPS2
               ZAHYO(J3JJ) = Y - EPS2

               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

               THP = THP
     #              -TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
     #              -TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)
     #              -TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

               ZAHYO(I3II) = X - EPS2
               ZAHYO(J3JJ) = Y + EPS2
               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

               THP = THP
     #              -TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)
     #              -TERSOFF_I(J,ZAHYO,LIST,KYORI,CUTOFF)
     #              -TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

               DATA = THP / ( 8.0D0 * EPS2 * EPS2 )

               ZAHYO(I3II) = X
               ZAHYO(J3JJ) = Y

               CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
               CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

               CALL ADD_DIFF2( DATA , I3II , J3JJ , DIFF2 ,DIFF2_DATA)


1420     CONTINUE
1430  CONTINUE

      RETURN
1499  END

1501  SUBROUTINE MAKE_LIST_IJ( I,J,LIST_IJ,LIST )
      INCLUDE 'PARAMETER'

      INTEGER I,J
      INTEGER LIST_IJ(0:MAX_LIST_IJ)
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)

      INTEGER L,M
```

```
      LIST_IJ(0) = 0

      DO 1540 L = 1 , LIST(0,I)
         DO 1530 H = 1 , LIST(0,J)
            IF ( LIST(L+MAX_LIST_N,I)
     #           .NE. LIST(H+MAX_LIST_N,J)) THEN
               GOTO 1530
            ENDIF

            IF ( LIST_IJ(0) .EQ. MAX_LIST_IJ )  THEN
               WRITE(*,*) 'MAX_LIST_IJ is overflow.'
            ENDIF

            LIST_IJ(0) = LIST_IJ(0) + 1
            LIST_IJ(LIST_IJ(0)) = LIST(L+MAX_LIST_N,I)

1530     CONTINUE
1540  CONTINUE

      RETURN
1599  END

1601  REAL*8 FUNCTION TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)
      INCLUDE 'PARAMETER'
      INTEGER LIST_IJ(0:MAX_LIST_IJ)
      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)
      REAL*8  TERSOFF_I

      INTEGER I,L

      TERSOFF_LIJ = 0.0D0

      DO 1610 L = 1 , LIST_IJ(0)

         I = LIST_IJ(L)

         TERSOFF_LIJ = TERSOFF_LIJ +
     #        TERSOFF_I(I,ZAHYO,LIST,KYORI,CUTOFF)

1610   CONTINUE

      RETURN

1699  END

1701  SUBROUTINE CALC_DIFF2_3
     #     (I,J,ZAHYO,LIST,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

      INCLUDE 'PARAMETER'
      INTEGER I,J
      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      INTEGER LIST_IJ(0:MAX_LIST_IJ)

      REAL*8  ZAHYO(MAX_ATOM3)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  TERSOFF_LIJ

      INTEGER I3 , J3
      INTEGER II,JJ
      INTEGER I3II,J3JJ
      REAL*8 X,Y

      REAL*8 THP
      REAL*8 DATA

      CALL MAKE_LIST_IJ( I,J,LIST_IJ,LIST )

      I3 = I*3 - 3
      J3 = J*3 - 3

      DO 1730 II = 1 , 3
         DO 1720 JJ = 1 , 3
            I3II = I3 + II
            J3JJ = J3 + JJ

            X = ZAHYO(I3II)
            Y = ZAHYO(J3JJ)

            ZAHYO(I3II) = X + EPS2
            ZAHYO(J3JJ) = Y + EPS2
            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)
```

```
            THP = TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

            ZAHYO(I3II) = X - EPS2
            ZAHYO(J3JJ) = Y - EPS2
            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

            THP = THP
     #           +TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

            ZAHYO(I3II) = X + EPS2
            ZAHYO(J3JJ) = Y - EPS2
            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

            THP = THP
     #           -TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

            ZAHYO(I3II) = X - EPS2
            ZAHYO(J3JJ) = Y + EPS2
            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

            THP = THP
     #           -TERSOFF_LIJ(LIST_IJ,ZAHYO,LIST,KYORI,CUTOFF)

            DATA = THP / ( 8.0D0 * EPS2 * EPS2 )

            ZAHYO(I3II) = X
            ZAHYO(J3JJ) = Y

            CALL RECALC(I,ZAHYO,LIST,KYORI,CUTOFF)
            CALL RECALC(J,ZAHYO,LIST,KYORI,CUTOFF)

            CALL ADD_DIFF2( DATA , I3II , J3JJ , DIFF2 ,DIFF2_DATA)

 1720    CONTINUE
 1730 CONTINUE

      RETURN
 1799 END

 1801 SUBROUTINE CONJGRAD(N,ZAHYO,DIFF1,DIFF2,DIFF2_DATA)
      INCLUDE 'PARAHETER'
      INTEGER N
      REAL*8 ZAHYO(HAX_ATOH3)
      REAL*8 DIFF1(HAX_ATOH3)
      INTEGER DIFF2(0:HAX_DIFF2_LIST,HAX_ATOH3)
      REAL*8  DIFF2_DATA(HAX_DIFF2_LIST,HAX_ATOH3)

      INTEGER N3

      REAL*8 P(HAX_ATOH3)
      REAL*8 R(HAX_ATOH3)
      REAL*8 AP(HAX_ATOH3)
      REAL*8 X(HAX_ATOH3)

      REAL*8 NORH_R2,NORH_R2_

      REAL*8 PAP

      REAL*8 A

      REAL*8 C

      INTEGER I,J,K
      INTEGER L

      N3 = N*3
      NORH_R2 = 0.0D0

      DO 1810 I = 1 , N3
         X(I) = 0.0D0
         R(I) = DIFF1(I)
         P(I) = DIFF1(I)
         NORH_R2 = NORH_R2 + R(I) * R(I)
 1810 CONTINUE

C      WRITE(*,*) 0,NORH_R2
      DO 1860 K = 1 , N3

         IF ( NORH_R2 .LT. 1.0D-6 ) THEN
            GOTO 1865
         ENDIF

         DO 1820 I = 1 , N3
            AP(I) = 0.0D0
            DO 1815 L = 1 , DIFF2(0,I)
               J = DIFF2(L,I)
               AP(I) = AP(I) + P(J) * DIFF2_DATA(L,I)

 1815       CONTINUE
```

```
 1820    CONTINUE

         PAP = 0.0D0
         DO 1830 I = 1 , N3
            PAP = PAP + P(I) * AP(I)
 1830    CONTINUE

         A = NORM_R2 / PAP
C         WRITE(*,*) 'A=',A

         NORM_R2_ = 0.0D0

         DO 1840 I = 1 , N3
            X(I) = X(I) + A * P(I)
            R(I) = R(I) - A * AP(I)
            NORM_R2_ = NORM_R2_ + R(I) * R(I)
 1840    CONTINUE

C         WRITE(*,*) K,NORM_R2_

         C = NORM_R2_ / NORM_R2
         IF ( C .GT. 1.0D0 ) THEN
            GOTO 1865
         ENDIF


         NORM_R2 = NORM_R2_
         DO 1850 I = 1 , N3
            P(I) = R(I) + C * P(I)
 1850    CONTINUE

 1860 CONTINUE

 1865 DO 1870 I = 1 , N3
         ZAHYO(I) = ZAHYO(I) + X(I)
 1870 CONTINUE

      RETURN
 1899 END

 1901 SUBROUTINE MAKE_LIST_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,IDX
     #,LIST,LIST2,AN)
      INCLUDE 'PARAMETER'
      INTEGER N,N3

      REAL*8  ZAHYO(MAX_ATOM3)
      CHARACTER*8  AN(MAX_ATOM)
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)
      INTEGER IDX

      INTEGER LIST(0:MAX_LIST_N2,MAX_ATOM)
      INTEGER LIST2(0:MAX_LIST2_N,MAX_ATOM)

      REAL*8  MAX_X,MAX_Y,MAX_Z
      REAL*8  MIN_X,MIN_Y,MIN_Z

      INTEGER I,K,J
      INTEGER E,F,G
      INTEGER L,M

      INTEGER I3

      INTEGER MESH_X,MESH_Y,MESH_Z
      INTEGER TX,TY,TZ
      INTEGER TI,TJ,TI3,TJ3

      INTEGER MESH_V(0:MAX_MESH_V_ATOM,
     #    0:MAX_MESH_V_X+1,0:MAX_MESH_V_Y+1,0:MAX_MESH_V_Z+1)

      REAL*8 THP_R,THP_CUTOFF

C      REAL*8 SIN


      N3 = N * 3

C     リスト配列の初期化
      DO 1905 I = 1 , MAX_ATOM
         LIST_VDW(0,I) = 0
 1905  CONTINUE

C     MESH 配列の初期化
      DO 1906 I = 0 , MAX_MESH_V_X+1
         DO 1906 J = 0 , MAX_MESH_V_Y+1
            DO 1906 K = 0 , MAX_MESH_V_Z+1
               MESH_V(0,I,J,K) = 0
 1906  CONTINUE

C     系の座標の最大値 - 最小値を検索する
      MAX_X = ZAHYO(1)
```

```
      HAX_Y = ZAHYO(2)
      HAX_Z = ZAHYO(3)
      HIN_X = ZAHYO(1)
      HIN_Y = ZAHYO(2)
      HIN_Z = ZAHYO(3)

      DO 1910 I = 4 , N3 , 3

         IF ( HAX_X .LT. ZAHYO(I) ) THEN
            HAX_X = ZAHYO(I)
         ENDIF
         IF ( HAX_Y .LT. ZAHYO(I+1) ) THEN
            HAX_Y = ZAHYO(I+1)
         ENDIF
         IF ( HAX_Z .LT. ZAHYO(I+2) ) THEN
            HAX_Z = ZAHYO(I+2)
         ENDIF
         IF ( HIN_X .GT. ZAHYO(I) ) THEN
            HIN_X = ZAHYO(I)
         ENDIF
         IF ( HIN_Y .GT. ZAHYO(I+1) ) THEN
            HIN_Y = ZAHYO(I+1)
         ENDIF
         IF ( HIN_Z .GT. ZAHYO(I+2) ) THEN
            HIN_Z = ZAHYO(I+2)
         ENDIF
 1910  CONTINUE

C     一度、大まかに原子をグルーピングする。
C     最大値最小値から、メッシュの個数を決める

      HESH_X = 1 + ( ( HAX_X - HIN_X ) / HESH_D_VDW )
      HESH_Y = 1 + ( ( HAX_Y - HIN_Y ) / HESH_D_VDW )
      HESH_Z = 1 + ( ( HAX_Z - HIN_Z ) / HESH_D_VDW )

C     HESH 用の配列 に収まらない場合 STOP
      IF ( HESH_X .LE. HAX_HESH_V_X ) THEN
         GOTO 1915
      ENDIF
      IF ( HESH_Y .LE. HAX_HESH_V_Y ) THEN
         GOTO 1915
      ENDIF
      IF ( HESH_Z .LE. HAX_HESH_V_Z ) THEN
         GOTO 1915
      ENDIF

      WRITE(*,*) 'HESH is overflow-vdw: HAX_V_HESH.',  HESH_X,HESH_Y,HESH_Z
      GOTO 1999


C     全ての原子を HESH に グルーピングする。

 1915  DO 1920 I = 1 , N
         I3 = I * 3 - 2
         TX = 1 + ( ( ZAHYO(I3) - HIN_X ) / HESH_D_VDW )
         TY = 1 + ( ( ZAHYO(I3+1) - HIN_Y ) / HESH_D_VDW )
         TZ = 1 + ( ( ZAHYO(I3+2) - HIN_Z ) / HESH_D_VDW )


C     HESH 配列があるれる場合 STOP
         IF ( HESH_V(0,TX,TY,TZ) .GE. HAX_HESH_V_ATOH ) THEN
            WRITE(*,*) 'HESH_V_ATOH is overflow: HAX_HESH_V_ATOH.'
            GOTO 1999
         ENDIF

         HESH_V(0,TX,TY,TZ) =  HESH_V(0,TX,TY,TZ) + 1
         HESH_V( HESH_V(0,TX,TY,TZ) , TX,TY,TZ) = I

 1920  CONTINUE


C     HESH のブロックの 近接内 (27 ブロック)
C     で 距離を計算し 近接 LIST を作成する
C     ついでに カットオフ関数の値も計算する

      DO 1943 I = 1 , HESH_X
         DO 1942 J = 1 , HESH_Y
            DO 1941 K = 1 , HESH_Z

               IF ( HESH_V(0,I,J,K) .EQ. 0 ) THEN
                  GOTO 1941
               ENDIF

               DO 1933 E = I - 1  , I + 1
                  DO 1932 F = J - 1 ,  J + 1
                     DO 1931 G = K - 1  , K + 1

                        IF ( HESH_V(0,E,F,G) .EQ. 0 ) THEN
                           GOTO 1931
                        ENDIF

                        DO 1925 L = 1 , HESH_V(0,I,J,K)
```

```
                        DO 1924 H = 1 , MESH_V(0,E,F,G)

                           TI = MESH_V(L,I,J,K)
                           TJ = MESH_V(H,E,F,G)

                           IF ( TI .LE. TJ ) THEN
                              GOTO 1924
                           ENDIF

                           TI3 = TI*3-2
                           TJ3 = TJ*3-2
C
C     原子間距離の計算
C
      THP_R =
     #     ( ( ZAHYO(TJ3  ) - ZAHYO(TI3  ) )**2 +
     #       ( ZAHYO(TJ3+1) - ZAHYO(TI3+1) )**2 +
     #       ( ZAHYO(TJ3+2) - ZAHYO(TI3+2) )**2  )**0.5d0

C
C     範囲外のとき飛ばす。

      IF ( THP_R .GT. (PRH_VCR2 + PRH_VCD2) ) THEN
      GOTO 1924
      ENDIF

      IF ( THP_R .LT. (PRH_VCR - PRH_VCD) ) THEN
      GOTO 1924
      ENDIF


C     同一分子内は 無視
      IF ( AN(TI) .EQ. AN(TJ) ) THEN
      GOTO 1924
      ENDIF


 1922 CONTINUE


C     LIST の配列があふれる時 STOP

      IF ( ( LIST_VDW(0,TI) .GE. MAX_LIST_VN ) .OR.
     #     ( LIST_VDW(0,TJ) .GE. MAX_LIST_VN ) ) THEN
         WRITE(*,*) 'LIST_VN is overflow: MAX_LIST_VN.'
         GOTO 1999
      ENDIF

      IDX = IDX + 1
      KYORI(IDX) = THP_R

      LIST_VDW(0,TI) = LIST_VDW(0,TI) + 1
      LIST_VDW( LIST_VDW(0,TI) ,TI ) = IDX
      LIST_VDW( LIST_VDW(0,TI) + MAX_LIST_VN ,TI ) = TJ

      LIST_VDW(0,TJ) = LIST_VDW(0,TJ) + 1
      LIST_VDW( LIST_VDW(0,TJ) ,TJ ) = IDX
      LIST_VDW( LIST_VDW(0,TJ) + MAX_LIST_VN ,TJ ) = TI

C
C     カットオフ関数を計算
C

      THP_CUTOFF = 0.0D0
      IF ( THP_R .GE. ( PRH_VCR - PRH_VCD) ) THEN
         THP_CUTOFF = 0.5D0 *
     #   ( 1.0D0 + SIN( PI * ( THP_R - PRH_VCR ) / (2.0D0 * PRH_VCD ) ))
      ENDIF
      IF ( THP_R .GE. ( PRH_VCR + PRH_VCD) ) THEN
         THP_CUTOFF = 1.0D0
      ENDIF
      IF ( THP_R .GE. ( PRH_VCR2 - PRH_VCD2) ) THEN
         THP_CUTOFF = 0.5D0 *
     #  ( 1.0D0 - SIN( PI * (THP_R - PRH_VCR2 )/(2.0D0 * PRH_VCD2 ) ))
      ENDIF
      IF ( THP_R .GE. ( PRH_VCR2 + PRH_VCD2) ) THEN
         THP_CUTOFF = 0.0D0
      ENDIF

      CUTOFF(IDX) = THP_CUTOFF

 1924  CONTINUE
 1925  CONTINUE


 1931  CONTINUE
 1932  CONTINUE
 1933  CONTINUE

 1941  CONTINUE
 1942  CONTINUE
 1943  CONTINUE
```

```
 1969  RETURN
 1999 END

C
C 原子 I に関した Van Der Waals ポテンシャル
C
 2001 REAL*8 FUNCTION VDW_I(I,ZAHYO,LIST_VDW,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I

      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST_VDW(0:MAX_LIST_VW2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      REAL*8  R,U
      REAL*8 VDW_FUNC

      INTEGER L,IDXJ

      U = 0.0D0

      DO 2010 L = 1  , LIST_VDW(0,I)
         IDXJ = LIST_VDW(L,I)
         R = KYORI(IDXJ)
C         U = U + CUTOFF(IDXJ) * VDW_FUNC(R)

         U = U + VDW_FUNC(R)


 2010 CONTINUE

      VDW_I = U
C      write(*,*) "VDW" , U , R

      RETURN

 2049 END

 2050 REAL*8 FUNCTION VDW_FUNC(R)
      INCLUDE 'PARAHETER'
      REAL*8 R,SR,SR2,SR6,SR12
      REAL*8 THP_CUTOFF
      REAL*8 SIN


      SR = 3.407D0 / R


      THP_CUTOFF = 0.0D0
      IF ( R .GE. ( PRM_VCR - PRM_VCD) ) THEN

      THP_CUTOFF = 0.5D0 *
     #  ( 1.0D0 + SIN( PI * ( R - PRM_VCR ) / (2.0D0 * PRM_VCD ) ))
      ENDIF
      IF ( R .GE. ( PRM_VCR + PRM_VCD) ) THEN
         THP_CUTOFF = 1.0D0
      ENDIF
      IF ( R .GE. ( PRM_VCR2 - PRM_VCD2) ) THEN
         THP_CUTOFF = 0.5D0 *
     #   ( 1.0D0 - SIN( PI * ( R - PRM_VCR2 ) / (2.0D0 * PRM_VCD2 ) ))
      ENDIF
      IF ( R .GE. ( PRM_VCR2 + PRM_VCD2) ) THEN
         THP_CUTOFF = 0.0D0
      ENDIF

SR2 = SR*SR
SR6 = SR2*SR2*SR2
       SR12 = SR6*SR6

      VDW_FUNC =
     #     4.0D0 * 0.002964D0 *
     #     ( SR12 - SR6 )
     #     * THP_CUTOFF


C      VDW_FUNC =
C     #     ( 4.0D0 * 0.002964D0 *
C     #     ( SR**12.0D0 - SR**6.0D0 ) + 0.000065290386570836 )
C     #     * THP_CUTOFF

C      VDW_FUNC =
C     #     ( 4.0D0 * 0.002964D0 *
C     #     ( SR**12.0D0 - SR**6.0D0 ) + -0.00000156405494 )
C     #     * THP_CUTOFF



      RETURN
 2099 END
```

```
 2101 SUBROUTINE CALC_DIFF1_VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF1)
      INCLUDE 'PARAHETER'
      INTEGER N,N3
      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST_VDW(0:HAX_LIST_VN2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      REAL*8  DIFF1(HAX_ATOH3)

      INTEGER I,ID3
      REAL*8  ZAHYO_ORG
      REAL*8 VDW_I

      REAL*8 THP

      N3 = N * 3

      DO 2110 I = 1 , N3

         ID3 = ( I + 2 ) / 3
         ZAHYO_ORG = ZAHYO(I)

         ZAHYO(I) = ZAHYO_ORG + EPS1
         CALL RECALC_VDW(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)
         THP = VDW_I(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)

         ZAHYO(I) = ZAHYO_ORG - EPS1
         CALL RECALC_VDW(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)
         THP = THP -
     #        VDW_I(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)

         ZAHYO(I) = ZAHYO_ORG
         CALL RECALC_VDW(ID3,ZAHYO,LIST_VDW,KYORI,CUTOFF)

         THP = THP / ( 2.0 * EPS1 )
C         WRITE(*,*) , I , THP
         DIFF1(I) = DIFF1(I) - THP
C         DIFF1(I) = - THP



 2110  CONTINUE

      RETURN
 2199  END
 2201 SUBROUTINE RECALC_VDW(I,ZAHYO,LIST_VDW,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I,I3
      REAL*8  ZAHYO(HAX_ATOH3)
      INTEGER LIST_VDW(0:HAX_LIST_VN2,HAX_ATOH)
      REAL*8  KYORI(HAX_DATA_IDX)
      REAL*8  CUTOFF(HAX_DATA_IDX)

      INTEGER J,J3
      INTEGER IDXJ
      INTEGER H

      REAL*8 THP_R
      REAL*8 THP_CUTOFF

C      REAL*8 SIN
      REAL*8 SQRT

      I3 = I*3-2

      DO 2210 H = 1 , LIST_VDW(0,I)
         IDXJ = LIST_VDW(H,I)
         J = LIST_VDW(H+HAX_LIST_VN,I)

         J3 = J*3-2
          THP_R =
     #    SQRT    ( ( ZAHYO(J3) - ZAHYO(I3) )**2 +
     #         ( ZAHYO(J3+1) - ZAHYO(I3+1) )**2 +
     #         ( ZAHYO(J3+2) - ZAHYO(I3+2) )**2  )


         KYORI(IDXJ) = THP_R

         THP_CUTOFF = 1.0D0
C         IF ( THP_R .GE. ( PRH_CR - PRH_CD) ) THEN
C         THP_CUTOFF = 0.5D0 *
C     #   ( 1.0D0 - SIN( PI * ( THP_R - PRH_CR ) / (2.0D0 * PRH_CD ) ))
C         ENDIF
C         IF ( THP_R .GE. ( PRH_CR + PRH_CD) ) THEN
C            THP_CUTOFF = 0.0D0
C         ENDIF


         CUTOFF(IDXJ) = THP_CUTOFF
```

```
 2210 CONTINUE

      RETURN
 2299 END

 2301  REAL*8 FUNCTION VDW(N,ZAHYO,LIST_VDW,KYORI,CUTOFF)
      INCLUDE 'PARAHETER'
      INTEGER I
      INTEGER N
      REAL*8  ZAHYO(MAX_ATOM3)
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)
      REAL*8  VDW_I

      VDW = 0.0D0

      DO 2310 I = 1 , N
         VDW = VDW +
     #        VDW_I(I,ZAHYO,LIST_VDW,KYORI,CUTOFF)

 2310  CONTINUE

      VDW = VDW / 2.0D0

      RETURN
 2399 END

 2401 SUBROUTINE CALC_DIFF2_VDW
     #(N,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
      INCLUDE 'PARAHETER'

      INTEGER N,N3
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)

      REAL*8  ZAHYO(MAX_ATOM3)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)

      INTEGER I,J,K
      INTEGER L

      N3 = N * 3

C      DO 2410 I = 1 , MAX_ATOM3
C         DIFF2(0,I) = 0
C 2410 CONTINUE

      DO 2450 I = 1 , N

C     隣合う原子の座標系
         DO 2445 L = 1 , LIST_VDW(0,I)
            J = LIST_VDW(L+MAX_LIST_VN,I)

            CALL CALC_DIFF2_VDW_1
     #           (I,J,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF2,DIFF2_DATA)
 2445    CONTINUE


 2450 CONTINUE

      RETURN
 2499 END

 2501 SUBROUTINE CALC_DIFF2_VDW_1
     #     (I,J,ZAHYO,LIST_VDW,KYORI,CUTOFF,DIFF2,DIFF2_DATA)

      INCLUDE 'PARAHETER'
      INTEGER I,J
      INTEGER LIST_VDW(0:MAX_LIST_VN2,MAX_ATOM)

      REAL*8  ZAHYO(MAX_ATOM3)
      REAL*8  KYORI(MAX_DATA_IDX)
      REAL*8  CUTOFF(MAX_DATA_IDX)

      INTEGER DIFF2(0:MAX_DIFF2_LIST,MAX_ATOM3)
      REAL*8  DIFF2_DATA(MAX_DIFF2_LIST,MAX_ATOM3)

      REAL*8  VDW_FUNC

      INTEGER I3 , J3
      INTEGER II,JJ
      INTEGER I3II,J3JJ
      REAL*8 X,Y

      REAL*8 THP
      REAL*8 DATA

      REAL*8 R,DIST
```

```
      REAL*8 GI(3)
      REAL*8 GJ(3)

      I3 = I*3 - 3
      J3 = J*3 - 3

      GI(1) = ZAHYO(I3 + 1)
      GI(2) = ZAHYO(I3 + 2)
      GI(3) = ZAHYO(I3 + 3)

      GJ(1) = ZAHYO(J3 + 1)
      GJ(2) = ZAHYO(J3 + 2)
      GJ(3) = ZAHYO(J3 + 3)


      DIST = CALC_R(GI,GJ)

      DO 2530 II = 1 , 3
         DO 2520 JJ = 1 , 3


               X = GI(II)
               Y = GJ(JJ)

               GI(II) = X + EPS2
               GJ(JJ) = Y + EPS2
               R = CALC_R(GI,GJ)

               THP = VDW_FUNC(R)

               GI(II) = X - EPS2
               GJ(JJ) = Y - EPS2
               R = CALC_R(GI,GJ)


               THP = THP +  VDW_FUNC(R)


               GI(II) = X + EPS2
               GJ(JJ) = Y - EPS2
               R = CALC_R(GI,GJ)

               THP = THP -  VDW_FUNC(R)



               GI(II) = X - EPS2
               GJ(JJ) = Y + EPS2
               R = CALC_R(GI,GJ)

               THP = THP - VDW_FUNC(R)

               DATA = THP / ( 8.0D0 * EPS2 * EPS2 )


               IF ( R .LT. (PRH_VCR2+PRH_VCD2) ) THEN
              CALL ADD2_DIFF2( DATA , I3+II , J3+JJ , DIFF2 ,DIFF2_DATA)
               ELSE
              CALL ADD_DIFF2( DATA , I3+II , J3+JJ , DIFF2 ,DIFF2_DATA)
               ENDIF
               GI(II) = X
               GJ(JJ) = Y


2520    CONTINUE
2530 CONTINUE

      RETURN
2599 END


2601 REAL*8 FUNCTION CALC_R(GI,GJ)

      REAL*8 GI(3)
      REAL*8 GJ(3)
      REAL*8 SQRT


      CALC_R = SQRT(
     #     (GI(1) - GJ(1))**2+
     #     (GI(2) - GJ(2))**2+
     #     (GI(3) - GJ(3))**2 )



      RETURN
2699 END


3401 SUBROUTINE INNERROTATE(N,ZAHYO,AN,DEG)
```

```
      INCLUDE 'PARAHETER'
      INTEGER N

C     loop variable
      INTEGER I,DEG

C     炭素原子の座標用配列

      CHARACTER*8 AN(HAX_ATOH)
      REAL*8 ZAHYO(HAX_ATOH3)

      REAL*8 DEGREE

      REAL*8 SIN,COS

      REAL*8 X,Y

      REAL*8 A11,A12,A21,A22

      A11 = COS(2.0*PI * DEG / 360.0)
      A12 = SIN(2.0*PI * DEG / 360.0)
      A21 = -A12
      A22 = A11

      DO 3450 I = 1 , N
         IF ( AN(I) .EQ. 'C' ) THEN

            X = ZAHYO(I*3-2)
            Y = ZAHYO(I*3-1)

            ZAHYO(I*3-2) = A11*X + A12*Y
            ZAHYO(I*3-1) = A21*X + A22*Y

         ENDIF

 3450 CONTINUE

      RETURN

 3499 END

 3501 SUBROUTINE INNERSLIDE(N,ZAHYO,AN,SL)

      INCLUDE 'PARAHETER'

C     loop variable
      INTEGER I,SL
      INTEGER N

C     炭素原子の座標用配列

      CHARACTER*8 AN(HAX_ATOH)
      REAL*8 ZAHYO(HAX_ATOH3)


      REAL*8 SLIDE

      SLIDE = SL / 1000.0

      DO 3550 I = 1 , N
         IF ( AN(I) .EQ. 'C1' ) THEN
            ZAHYO(I*3) = ZAHYO(I*3) + 1.0D0 * SLIDE
         ENDIF

 3550 CONTINUE

      RETURN
      END
```

B.3 経験ポテンシャルのパラメータファイル

ここで示すファイル 経験ポテンシャルで用いられる係数とパラメータとして定義する物である。 fortran のプログラムの各 function や subroutine から参照される。

計算において変更する必要があるのは

```
c       プログラムで用いる炭素原子の最大個数
c
        INTEGER MAX_ATOM
        PARAMETER (MAX_ATOM = 5000)
```

もちいる分子の大きさを考慮して、 1 で 役 1.5Å の大きさと考えておくとよい。

```
c       list を作るときの メッシュの最大値
        INTEGER MAX_MESH_X
        PARAMETER (MAX_MESH_X = 25)
        INTEGER MAX_MESH_Y
        PARAMETER (MAX_MESH_Y = 25)
        INTEGER MAX_MESH_Z
        PARAMETER (MAX_MESH_Z = 50)
```

共役勾配法を用いる時は 2次微分係数のリスト用配列をとる必要がある。多くのメモリーを消費するので注意。

```
        INTEGER MAX_DIFF2_LIST
C        PARAMETER (MAX_DIFF2_LIST = 400)
        PARAMETER (MAX_DIFF2_LIST = 1)
```

```
C     変数型 は STRICT
C      IHPLICIT LOGICAL ( A-Z )
IHPLICIT REAL*8 ( A-Z )


c     Tesoff ポテンシャルの
c     パラメータ ( 炭素原子 )

      REAL*8 PRH_EA
      REAL*8 PRH_EB

      REAL*8 PRH_LUHBDA1
      REAL*8 PRH_LUHBDA2
      REAL*8 PRH_LUHBDA3

      REAL*8 PRH_ALPHA
      REAL*8 PRH_DELTA
      REAL*8 PRH_ETA

      REAL*8 PRH_A
      REAL*8 PRH_C
      REAL*8 PRH_D
      REAL*8 PRH_H

      REAL*8 PRH_CR
      REAL*8 PRH_CD

      PARAMETER (PRH_EA =  1393.6D0)
      PARAMETER (PRH_EB =  -346.74D0)

C normal
```

```
C      PARAHETER (PRH_LUHBDA1 = 3.4879D0)
C      PARAHETER (PRH_LUHBDA2 = 2.2119D0)

C Graphite saigen
       PARAHETER (PRH_LUHBDA1 = 3.6012997D0)
       PARAHETER (PRH_LUHBDA2 = 2.28381399D0)

C C60 saigen

C      PARAHETER (PRH_LUHBDA1 = 3.6219D0)
C      PARAHETER (PRH_LUHBDA2 = 2.2969D0)


       PARAHETER (PRH_LUHBDA3 = 0.0D0)

       PARAHETER (PRH_ALPHA = 0.0D0)
       PARAHETER (PRH_DELTA = 0.687276D0)
       PARAHETER (PRH_ETA   = 0.72752D0)

       PARAHETER (PRH_A  =  1.5724D-7)
       PARAHETER (PRH_C  =  38049.0D0)
       PARAHETER (PRH_D  =  4.3484)
C      PARAHETER (PRH_D  =  4.384)
       PARAHETER (PRH_H  = -0.57058)

       PARAHETER (PRH_CR  = 1.95D0)
       PARAHETER (PRH_CD  = 0.15D0)


c      円周率π
       REAL*8 PI
       PARAHETER (PI  = 3.14159265358979323846D0)

c      プログラムで用いる炭素原子の最大個数
c
       INTEGER HAX_ATOH
       PARAHETER (HAX_ATOH = 5000)

c      炭素原子の最大個数の３倍
c
       INTEGER HAX_ATOH3
       PARAHETER (HAX_ATOH3 = HAX_ATOH * 3)

c      list を作るときの メッシュの最大値
       INTEGER HAX_HESH_X
       PARAHETER (HAX_HESH_X = 25)
       INTEGER HAX_HESH_Y
       PARAHETER (HAX_HESH_Y = 25)
       INTEGER HAX_HESH_Z
       PARAHETER (HAX_HESH_Z = 50)


c      一つのメッシュの最大個数
       INTEGER HAX_HESH_ATOH
       PARAHETER (HAX_HESH_ATOH = 50)

c      一つの list の最大個数
       INTEGER HAX_LIST_N
       PARAHETER (HAX_LIST_N = 20)

c      一つの list の最大個数 の２倍
       INTEGER HAX_LIST_N2
       PARAHETER (HAX_LIST_N2 = HAX_LIST_N * 2)

c      一つの list2 の最大個数
       INTEGER HAX_LIST2_N
C       PARAHETER (HAX_LIST2_N = 40)
       PARAHETER (HAX_LIST2_N = 20)



       INTEGER HAX_DIFF2_LIST
C       PARAHETER (HAX_DIFF2_LIST = 400)
       PARAHETER (HAX_DIFF2_LIST = 1)

       INTEGER HAX_LIST_IJ
       PARAHETER (HAX_LIST_IJ = 10)

c      距離 カットオフのデータ領域の大きさ
       INTEGER HAX_DATA_IDX
       PARAHETER (HAX_DATA_IDX = 500000)

c      メッシュ間隔
       INTEGER HESH_D
       PARAHETER ( HESH_D = ( PRH_CR + PRH_CD ) * 1.01D0 )

       PARAHETER (EPS1  = 0.000001D0)
       REAL*8 EPS2
       PARAHETER (EPS2  = 0.00001D0)

       REAL*8 TIHE_DIV
       PARAHETER (TIHE_DIV = 2.5D-15)
```

```
C 2.5
      REAL*8 Na
      PARAMETER (Na = 6.022045D+23)

      REAL*8 HassC
      PARAMETER (HassC = 12.0D0 * 1.6605655D-27)

      REAL*8 CONV_VELO
      PARAMETER (CONV_VELO =
     # 1.6021892D-19/HassC*TIME_DIV*TIME_DIV*1.0D+20)

      REAL*8 PRM_VCR
      REAL*8 PRM_VCD

      PARAMETER (PRM_VCR  = 2.1D0)
      PARAMETER (PRM_VCD  = 0.1D0)

      REAL*8 PRM_VCR2
      REAL*8 PRM_VCD2

      PARAMETER (PRM_VCR2  = 17.5D0)
      PARAMETER (PRM_VCD2  = 1.0D0)

      REAL*8 MESH_D_VDW
      PARAMETER ( MESH_D_VDW = ( PRM_VCR2 + PRM_VCD2 ) * 1.01D0 )

c     vdw の list を作るときの メッシュの最大値
      INTEGER MAX_MESH_V_X
      PARAMETER (MAX_MESH_V_X = 7)
      INTEGER MAX_MESH_V_Y
      PARAMETER (MAX_MESH_V_Y = 7)
      INTEGER MAX_MESH_V_Z
      PARAMETER (MAX_MESH_V_Z = 20)

c     vdw 一つのメッシュの最大個数
      INTEGER MAX_MESH_V_ATOM
      PARAMETER (MAX_MESH_V_ATOM = 2100)

c     一つの list の最大個数
      INTEGER MAX_LIST_VN
      PARAMETER (MAX_LIST_VN = 2100)

c     一つの list の最大個数 の２倍
      INTEGER MAX_LIST_VN2
      PARAMETER (MAX_LIST_VN2 = MAX_LIST_VN * 2)
```

付録 C

付録 著者の学外における発表実績

カーボンナノチューブの面間相互作用
松尾 竜馬 齋藤 理一郎 木村 忠正
1999 年 9 月 岩手大学 日本物理学会 秋の分科会

多層カーボンナノチューブの安定構造
松尾 竜馬 齋藤 理一郎 木村 忠正
2000 年 1 月 岡崎カンファレンスセンター 第 18 回フラーレンジンポジウム