

1997 年度 修士論文

カーボンナノチューブの
フォノン分散関係とラマン強度

9630033

竹谷 隆夫

電気通信大学 大学院 電子工学専攻 電子デバイス工学講座

指導教官 齋藤 理一郎 助教授

木村 忠正 教授

提出日 平成 9 年 2 月 4 日

謝辞

本研究を進めるにあたって多大な御指導、御助言を頂きました本学電子工学科齊藤理一郎助教授に心より御礼を申し上げます。

また、本研究に数々の有益な御助言を頂いた木村忠正教授、湯郷成美助教授、一色秀夫助手に深謝を申しあげます。

また、研究活動をともにし、多くの援助をいただいた八木将志氏、松尾竜馬氏、グェンドゥック ミン氏に深謝いたします。

そして、数々の御援助、御助言をしていただいた出島徹氏、池田典昭氏、菅野仙子氏、高橋雅也氏、棚谷公彦氏、中村伸之氏はじめ木村・齋藤・湯郷研究室の大学院生、卒研生の方々に感謝します。

最後に、事務業務をして頂いた山本純子さんに感謝致します。

平成 10 年 2 月 4 日

著者

目次

1	序論	1
1.1	背景	1
1.2	カーボンナノチューブの歴史	2
1.3	カーボンナノチューブの分子構造	2
1.3.1	カーボンナノチューブの種類	2
1.3.2	カイラルベクトル、カイラル角 (螺旋度)	3
1.3.3	並進ベクトル	5
1.3.4	対称ベクトル R	5
1.4	カーボンナノチューブの電子物性	6
1.5	カーボンナノチューブのフォノン分散関係	7
1.6	単層カーボンナノチューブ (SWCN) のラマン強度の実験	7
1.7	目的	10
2	方法	11
2.1	フォノン分散関係を求めるための運動方程式	11
2.2	2次元グラファイトのフォノン分散関係	13
2.3	チューブのフォノン分散関係	16
2.3.1	zone-folding の近似の計算方法	16
2.3.2	3次元の力のテンソル	16
2.3.3	1D ナノチューブの円筒面効果のための力の定数のパラメーター の補正	18
2.4	カーボンナノチューブのラマン強度	21
2.4.1	ラマン散乱の原理	21
2.4.2	結合分極近似	23

3	結果・考察	26
3.1	フォノン分散関係	26
3.1.1	グラファイトのフォノン分散関係	26
3.1.2	チューブのフォノン分散関係	28
3.1.3	カーボンナノチューブの音速度の螺旋度依存性	30
3.2	カーボンナノチューブのラマン強度	31
3.2.1	カーボンナノチューブがランダムにある場合のラマン強度	31
3.2.2	カーボンナノチューブのラマン強度の角度依存性	41
3.2.3	フォノン状態密度とラマン強度	43
3.2.4	カーボンナノチューブのラマン強度の端の効果	44
4	結論	54
A	データ	57
A.1	ナノチューブの端ノ効果による軸方向のモードのデータ	57
A.2	データの場所	59
B	プログラムソース	60
B.1	円筒形の座標を求めるプログラムソース	60
B.2	nT ナノチューブの座標を求めるプログラム	66
B.3	2D グラファイトのフォノン分散関係を求めるプログラム	69
B.4	ナノチューブ分散関係を求めるプログラム	83
B.5	状態密度計算プログラム	101
B.6	立体角ラマン強度計算プログラム	103
B.7	立体角ラマン強度計算プログラム	103
B.8	端のある立体角ラマン強度計算プログラム	111
B.9	ラマン強度計算プログラム (角度依存性)	123

第 1 章

序論

この章では、章 1 で本研究に至るまでの背景を述べる。章 1.2、1.3 では、本研究の対象としているカーボンナノチューブの歴史と分子構造を述べる。また章 1.4 ではカーボンナノチューブの螺旋度の変化による電子物性を述べる。次いで、章 1.5 で、チューブのフォノン分散関係を求めるにあつた問題点を述べ、章 1.6 で単層カーボンナノチューブのラマン強度の実験について述べる。最後に、章 1.7 で本研究の目的を述べる。

1.1 背景

本研究の対象としているカーボンナノチューブは、グラファイト層を巻いて作られたナノスケールの物質である。その巻き方によってさまざまな螺旋度や半径を持つチューブができる。

最近レーザーアブレーションの方法を用い、A. M. Rao[2]らによって rope と呼ばれる螺旋度 (10,10) の単層カーボンナノチューブ (SWCN) からなる結晶が生成され、ラマン強度の実験の論文 [2] が報告がされた。この論文 [2] の中で、理論的解析として螺旋度のない、いわゆる armchair 型と呼ばれるナノチューブの解析しかしていない。また、H. Kataura[11]らによる螺旋度を持つ SWCN のラマン強度の実験も報告されており、螺旋度をもつ SWCN のラマン強度の理論的解析が必要とされている。

ナノチューブのラマン強度を計算するためには、まずフォノン分散関係を求めなければいけないが、その一つの方法として、R. A. Jishi[1]らによる zone-folding の近似があげられる。しかし、この近似 [1] は、ナノチューブの実際の x , y , z 方向の音響モードやナノチューブの太さが変化するいわゆるブリージングモードを正しく表現できないことが知られており、また一般の螺旋度を持つナノチューブのフォノン分散関係を

求めることが非常に困難であることが知られている。

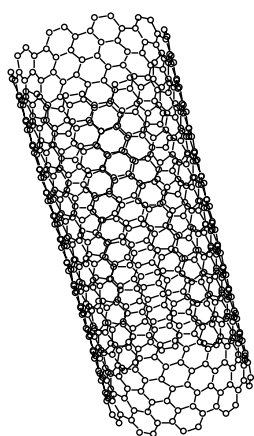
1.2 カーボンナノチューブの歴史

フラーレンは偶数個の炭素原子からなる閉殻構造を有する多面体クラスターである。 C_{500} 程度の巨大フラーレンは、すでに発見されていた。今まで発見されてきたこのような巨大フラーレンより更に大きな nano-scale サイズの炭素分子の集まりとして日本電気基礎研究所の飯島らのグループらによって 1991 年に発見されたものがあり、これがカーボンナノチューブとよばれるものである。飯島らは、希ガス下のアーク放電により生成するフラーレンを含んだススよりも、放電後に負電極の周囲に残る堆積物に注目した。この堆積物にはナノチューブのほとんどが 2 ~ 50 層からなる入れ子構造をとっていた。チューブの先端があたかも長いフラーレンの様に丸く閉じていたことである。このことはトポジカルな考察から、六員環のみならず五員環が炭素ネットワークを形成していることがわかる。この意味でチューブは巨大なフラーレンとみなすことができる。

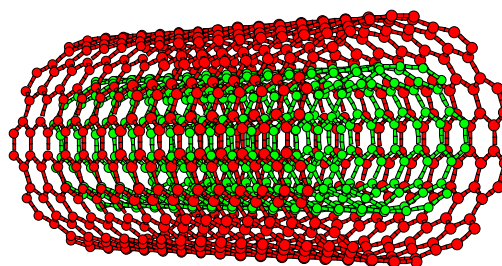
1.3 カーボンナノチューブの分子構造

1.3.1 カーボンナノチューブの種類

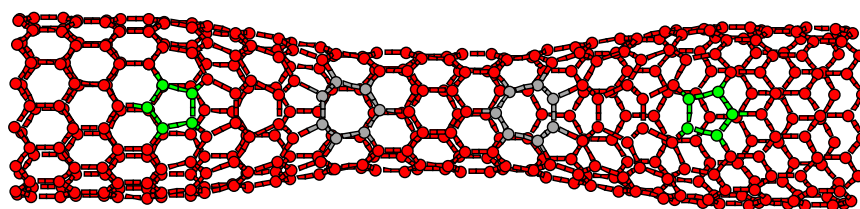
カーボンナノチューブは、フラーレンの拡大解釈されたものと考えられ、形状はグラファイト平面を丸めて円筒形にしたものである。その巻き方によってさまざまな半径、螺旋度を持つナノチューブができる。カーボンナノチューブの種類として図 1.1 の (a) 単層カーボンナノチューブ (SWCN) (六員環のみが存在)、(b) 入れ子構造 (六員環のみが存在)、(c) 太さの違う円筒形チューブをつないだものがあげられる (六員環だけでなく五、七員環が存在)。また、実験で生成されるナノチューブの多くは、(d) の様に、端にキャップ (六、五員環が存在) をもっている。



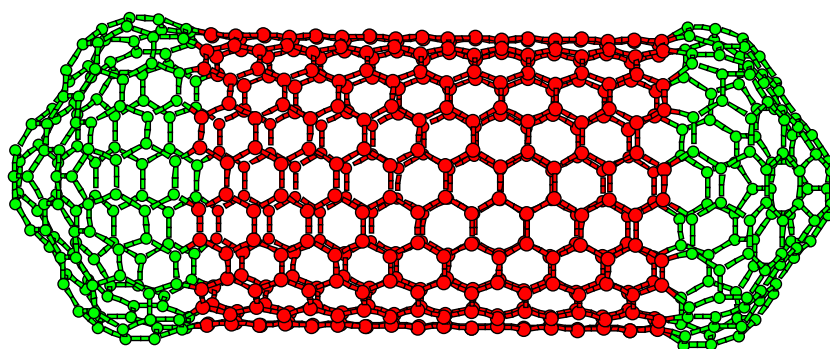
(a) 単層ナノチューブ



(b) 入れ子のナノチューブ



(c) 太さの違うナノチューブを組み合わせたナノチューブ



(d) キャップをもっているナノチューブ

図 1.1. カーボンナノチューブの種類

1.3.2 カイラルベクトル、カイラル角 (螺旋度)

図 1.2 に、カーボンナノチューブの展開図を示す。

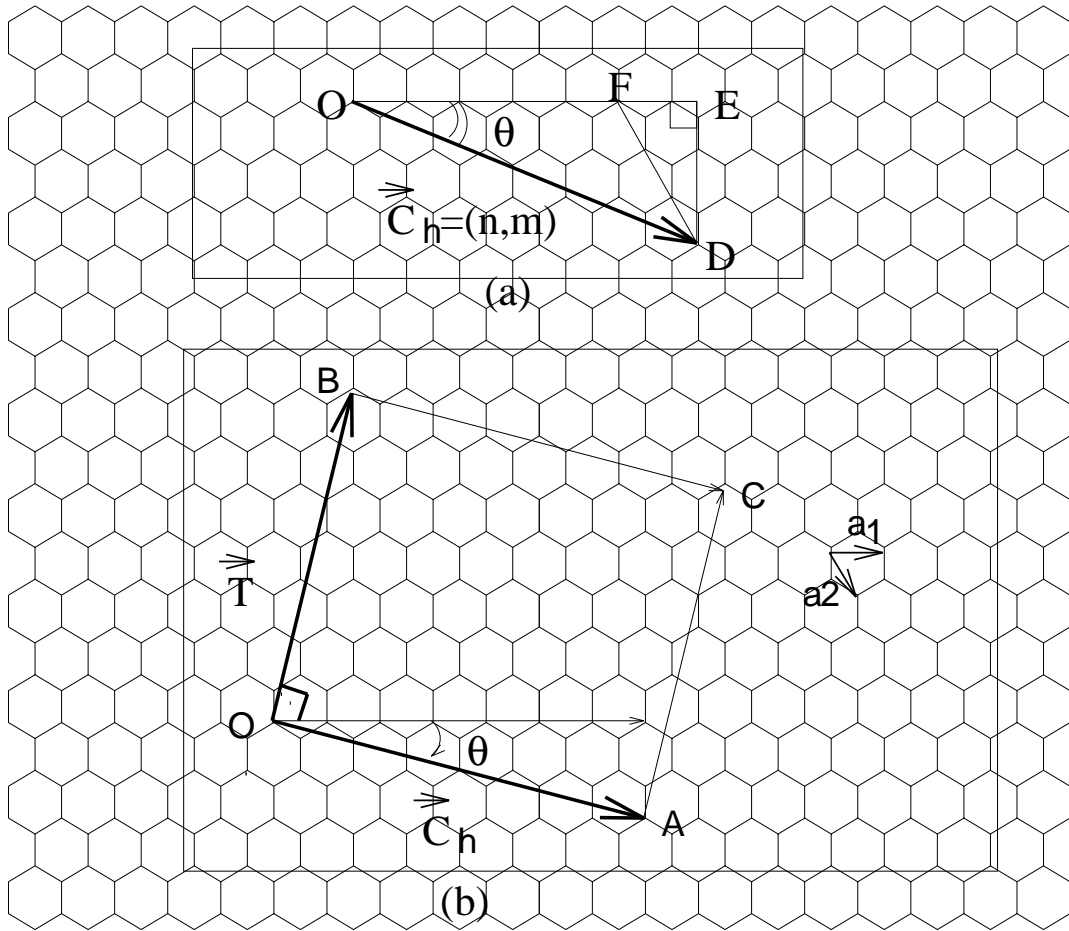


図 1.2. カーボンナノチューブの展開図

まず、チューブの構造を理解する上で必要なものとして、カイラルベクトルがあげられる。図 1.2に示す通りカイラルベクトルとは円筒面の展開図においてチューブの赤道（即ち円周）に相当するものである。OB と AC をつなげることによって円筒型チューブができる。またグラファイトの基本格子ベクトル a_1, a_2 を用いて

$$C_h = na_1 + ma_2 = (n, m), \quad (n, m \text{ は整数}, 0 < |m| < n) \quad (1.1)$$

で表される。 a_1, a_2 ベクトルの大きさは、炭素原子距離 a_{c-c} が 1.412 であることより、 $a = |a_1| = |a_2| = \sqrt{3}a_{c-c}$ である。またチューブの円周の長さ L 、即ち $|C_h|$ は図 1(a) より求められる。例えば $|C_h| = (n, m)$ とすると $OF = na, FD = am$ 、 $\angle EFD = \pi/3$ また、 $FE = am/2$ 、 $ED = \sqrt{3}am/2$ より、次式で表される。

$$|C_h| = L = \sqrt{OE^2 + ED^2} = a\sqrt{n^2 + m^2 + nm} \quad (1.2)$$

よってチューブの直径 d_t は $d_t = \frac{L}{\pi}$ で与えられる。

次に、 a_1 と $|C_h|$ のなす角をカイラル角 θ とよぶ。六角形の対称性より、 $\theta = \frac{\pi}{3}$ 以

上、 $\frac{2}{3}$ 以下の範囲で定義でき、図 1.2(b) を見て、 $\tan(\theta) = ED/OE$ より次式で表される。

$$\theta = \tan^{-1} \frac{\sqrt{3}m}{2n+m} \quad (1.3)$$

ここで、図 1.2(b) でのチューブの展開図上で OB と AC をくっつけることによって円筒型のチューブができる。この円筒型の中でも、カイラルベクトル $C_h = (n, 0)$ のものを zigzag 型、 $C_h = (n, n)$ のものを arm-chair 型とよぶ。またこの時のカイラル角 θ はそれぞれ ± 30 度、 0 度である。

1.3.3 並進ベクトル

図 1.2(b) で O から C_h に垂直な方向に伸ばしていき O と最初に等価な格子点を B とおく。OB を並進ベクトル T とよぶ。T は a_1 、 a_2 を用いて次式で表される。

$$T = t_1 a_1 + t_2 a_2 = (t_1, t_2) \text{ (ただし } t_1, t_2 \text{ は互いに素)} \quad (1.4)$$

ここで、 t_1, t_2 は C_h と T は垂直なことと内積の定理と、ユークリッドの互除法をもちいて以下のように表される。

$$t_1 = \frac{2m+n}{d_R}, \quad t_2 = -\frac{2n+m}{d_R} \quad (d_R \text{ は、} (2m+n) \text{ と } (2n+m) \text{ の最大公約数}), \quad (1.5)$$

で表される。

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

$$N = 2 \frac{(n^2 + m^2 + nm)}{d_R} \quad (1.6)$$

これよりチューブのユニットセル内の炭素原子の数は、オイラーの定理より、 $2N$ となる。

1.3.4 対称ベクトル R

図 1(b) の格子点 O から出発してユニットセル内の N 個の格子点 (原子) をとるベクトルを対象ベクトル R とよぶ。R は次式で表される。

$$R = p a_1 + q a_2 = (p, q), \text{ (ただし } p, q \text{ は互いに素)} \quad (1.7)$$

ここで、 p, q は t_1, t_2 を用いて次式で定義できる。

$$t_1q - t_2p = 1, \quad (0 < mp - nq < N) \tag{1.8}$$

この対称ベクトル R は展開図からコンピュータ上でチューブを設計するにあたって役に立つ。

1.4 カーボンナノチューブの電子物性

図 1.3 に螺旋度 $(10,10)$ のナノチューブの半径の付近にあるナノチューブを示す。ここで (n, m) は、ナノチューブの螺旋度 (カイラルベクトル) を示し、その下は、各螺旋度にある単位胞内の原子数を示している。また、各螺旋度において白丸は金属的性質、黒丸は半導体的性質を示す。

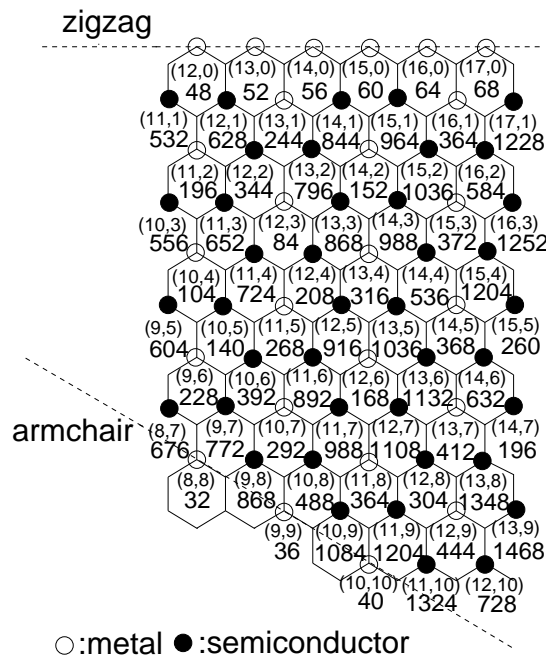


図 1.3. チューブの電子物性

図 1.3 の様に螺旋度により、電子状態が金属的性質になったり、半導体的性質を持つ。またその螺旋構造のピッチと、チューブの径により、そのバンドギャップの大きさを制御できる。

1.5 カーボンナノチューブのフォノン分散関係

まず、カーボンナノチューブのラマン強度を求めるためにはフォノン分散関係を求める。ナノチューブのフォノン分散関係を求める方法の一つとして、R. A. Jishi[1] による、二次元グラファイトのブリルアンゾーンを丸めて一次元ブリルアンゾーンをもつカーボンナノチューブの分散関係を求める zone-folding の近似 [1] があげられる。しかしながら、この近似を使うといくつかのモードで補正が必要で、ナノチューブは 1 p 次元の波数方向を持っているのに、波数の 2 乗に比例する音響モードが表れるなどの正確なナノチューブの分散関係を与えない。例えば、図 1.4(a) の左図に見られるにグラファイトの $k=0$ における音響モードの一つである LA モード (out-of-plane) が丸めてチューブにした場合、図 1.4の (a) 右図の様なチューブのラマン活性モードの一つである動径方向に伸縮するブリージングモードになり、周波数 $\omega \neq 0$ となってしまう。

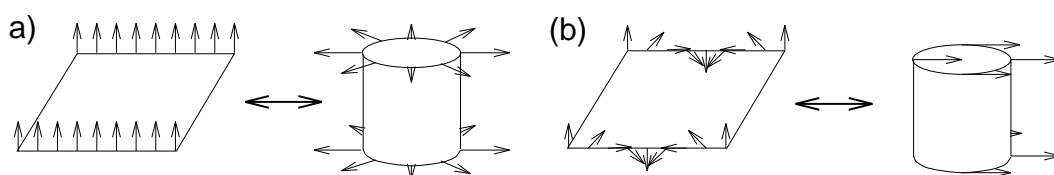


図 1.4. 2D グラファイトの LA モードとチューブの LA モードの違い

一方では、三次元空間でのチューブの音響モードとして、チューブの軸方向垂直な x, y 方向と、チューブの軸方向の z 方向のモードを一般的に考える。しかし、例えば、チューブの軸方向と垂直な方向の音響モード (x, y 方向) は、2次元のグラファイトのどのモードとも一致しない。

そこで、上で述べた様な違い避け、より現実的な結果を得るために、チューブの座標を直接使い 3 次元の力のテンソルを定義することによって、フォノン分散関係を求める必要がある。

1.6 単層カーボンナノチューブ (SWCN) のラマン強度の実験

A. M. Rao[2] らによって螺旋度 (10,10) の単層カーボンナノチューブ (SWCN) のラマン強度の実験の論文 [2] が報告された (図 1.5)。しかしながら、この論文の中では、armchair 型といわれる螺旋度のないナノチューブの解析しかしていない。ラマン強度の実験において試料であるロープ状単層カーボンナノチューブ (SWCN) は、金属の触

媒入りカーボンロッド用いて、レーザー蒸発法で得ることができる。A. Thess[13]らのグループは、NiCo 螺触媒入りカーボンロッドを使い、電気炉内ダブルレーザー蒸発法で、螺旋度 (10,10) の SWCN を非常に高い収率で得ることに成功した。一方では、H. Kataura [11]らによって螺旋度を持つ SWCN も得られている。彼らは NiCo の触媒入りカーボンロッドを使い、シングルレーザー蒸発法を用いて、These[?]らのグループのグループには、収率では及ばないものの、同じ直径であるが螺旋度には変化があるものが得られている。しかしながら、生成される SWCN の螺旋度や半径は非常に狭い分布にあるので成長温度などの成長条件に敏感である。例えば、カーボンロッドの重さに対して、触媒である Ni/Co を 1.2 % とし、カーボンロッドを 500Torr の Ar ガスフロー中で温度 1190 ° C に保ち生成したローブは直径が 1.0-1.4nm であるのに対して、触媒を Ph/Pd を 2.4 % とし 500Torr の Ar ガスフロー中で温度 1100 ° C の場合、生成したローブは直径が 0.8-1.0nm である。

図 1.5に A. M. Rao[2]による励起光源が 514.5nm である Ar⁺ レーザーを用いて、螺旋度 (10,10) の単相カーボンナノチューブのラマン強度の実験と、理論解析結果 [2]を示す。一番上がラマン強度実験であり、その下からそれぞれ順に螺旋度 (11,11)、(10,10)、(9,9)、(8,8) の同じ螺旋度を持つ半径の違うナノチューブの理論値を示している。

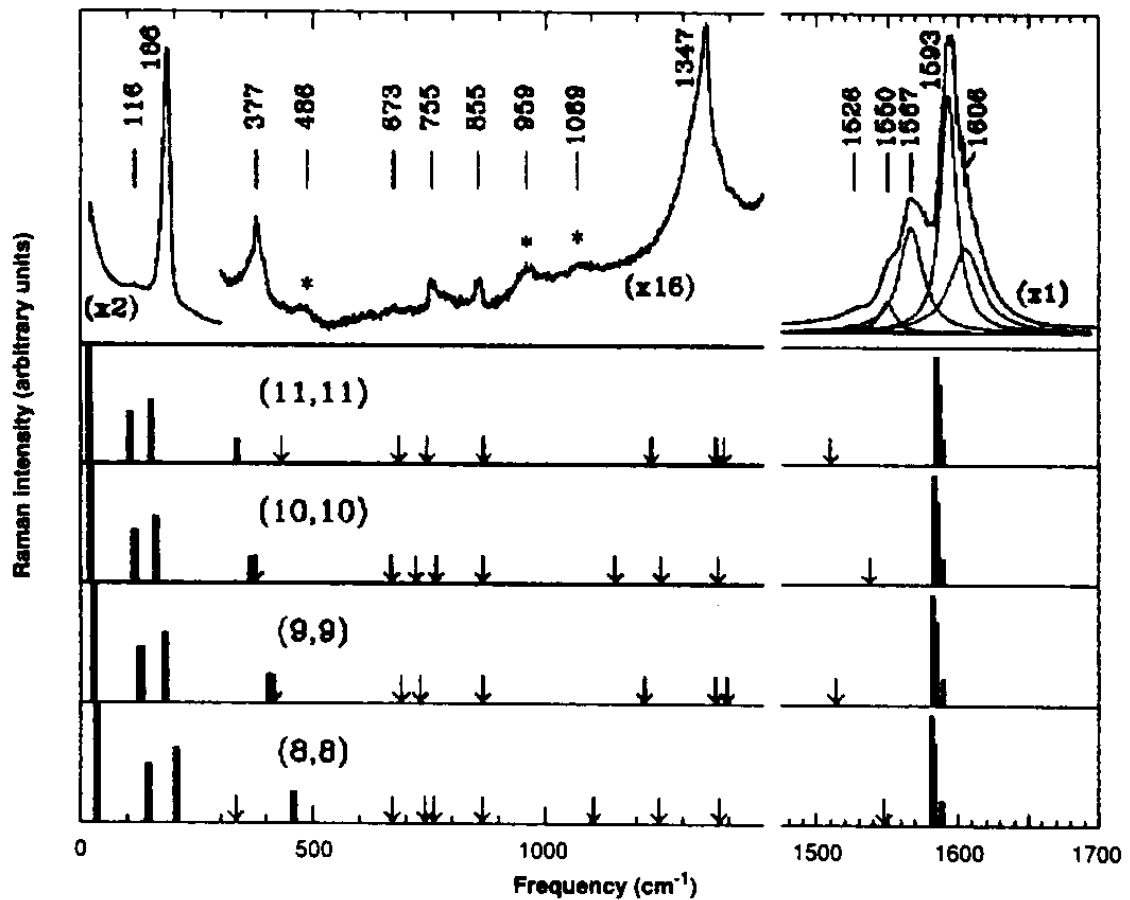


図 1.5.A. M. Rao[2] からのラマン強度の実験値 (一番上) と理論的解析

群論の予想より螺旋度があるもの、ないもののラマン活性モードはそれぞれ 16、15 個あることがわかっている。図 1.5 の実験値 [2] より、15 個のラマンスペクトルがあることがわかる。実験値において、 1600cm^{-1} 付近のピークは、グラファイトシートの振動モード対応するものである。高周波数領域にラマン活性周波数は図 1.5 を見てわかるように、半径依存性が見られない。

図 1.5 実験値で見られる、 1347cm^{-1} の強度であるが、図 1.5 の理論値 [2] と比べてわかるように理論値には表れていないが、これはナノチューブのキャップや、格子欠陥からくるのかもしれない。

一方、 186cm^{-1} の付近のモードは単相カーボンナノチューブ (SWCN) 固有のモードである A_{1g} (ブリージングモード) である (章 3.2.1 の図 3.11(c) 参照)。また、この A_{1g} は、カーボンロッド中の NiCo の濃度変える等、作成条件を変えると系統的にスペクトルが変化することが H. Kataura らによって報告 [11] されている。例えば、励起光源が 488nm である Ar^+ を用いた場合、 162 、 182cm^{-1} にラマンスペクトルが得られるが、励起光源が 514.5nm にすると 162cm^{-1} のスペクトルは消え、 182cm^{-1} のピーク

クは、 185cm^{-1} へシフト化する。さらに、他の発振器を使って A_{1g} モードのラマン強度を測定すると、わずかな波長の変化に対して、スペクトルは大幅な変化することが観測されると、報告している。これらの共鳴効果は SWCN の電子の状態密度のシャープな振動構造を反映して生じていると思われる。

図 1.5 の理論値 [2] で表れている、最も低い周波数にある E_{2g} モード (章 3.2.1 図 3.11(a)) であるが、 0cm^{-1} 付近のレイリー散乱のために実験値には表れていない。

1.7 目的

A. M. Rao による論文 [2] の中では、螺旋度のないいわゆる armchair 型と呼ばれるカーボンナノチューブの理論的解析しかしていない。本研究では、一般の螺旋度を持つカーボンナノチューブのフォノン分散関係において、より現実的な結果を得るために、チューブの座標より三次元の力のテンソルを定義し、分散関係を求める。また、得られたフォノン分散関係の 点の固有値、固有ベクトルを用いて結合分極近似 [3] を使い、ナノチューブのラマン強度を求め、その構造依存性について述べる。

第 2 章

方法

この章では計算方法を述べる。章 2.1 ではグラファイト、チューブのフォノン分散関係の求める。また、章 2.2 では、結合分極近似を使つてのチューブのラマン強度の求める。

2.1 フォノン分散関係を求めるための運動方程式

まず、フォノン分散関係を求めるために、ユニットセル内の N 個の炭素原子の運動方程式

$$M_i \ddot{\mathbf{u}}_i = \sum_j K^{(ij)} (\mathbf{u}_j - \mathbf{u}_i), \quad (i = 1, \dots, N), \quad (2.1)$$

を解く。ここで、 M_i は原子の質量、 $\mathbf{u}_j = (x_j, y_j, z_j)$ 、 $\mathbf{u}_i = (x_i, y_i, z_i)$ は i, j 番目のそれぞれの原子の位置座標、 $K^{(ij)}$ は i 番目の j 番目原子に対する 3×3 の力のテンソルである。また、式 (2.1) は i 番目の原子に対して第 1 近接から、第 n 近接までの j 番目の原子の和をとり、 n の数が多いほど、より現実的な分散関係を得ることができる。

周期的構造では、波数ベクトル k' を用いて \mathbf{u}_i をフーリエ変換 $\mathbf{u}_{\mathbf{k}'}$

$$\mathbf{u}_i = \frac{1}{\sqrt{N_\Omega}} \sum_{\mathbf{k}'} e^{-i(\mathbf{k}' \cdot \mathbf{R}_i - \omega t)} \mathbf{u}_{\mathbf{k}'}, \quad \text{または} \quad \mathbf{u}_{\mathbf{k}'} = \frac{1}{\sqrt{N_\Omega}} \sum_{\mathbf{R}_i} e^{i(\mathbf{k}' \cdot \mathbf{R}_i - \omega t)} \mathbf{u}_i, \quad (2.2)$$

ここで、 k' 、 N_Ω は、それぞれ第一ブリルアンゾーンの波数ベクトル、個体中のユニットセルの数であり、 \mathbf{R}_i は i 番目の原子の座標を示す。また、 \mathbf{u}_i を固有周波数 ω の関数として 2 階時間微分すると $\ddot{\mathbf{u}}_i = -\omega^2 \mathbf{u}_i$ となり、は式 (2.1) 次の式に変形される。

$$\left(\sum_j K^{(ij)} - M_i \omega^2 \right) \sum_{\mathbf{k}'} e^{-i\mathbf{k}' \cdot \mathbf{R}_i} \mathbf{u}_{\mathbf{k}'}^{(i)} = \sum_j K^{(ij)} \sum_{\mathbf{k}'} e^{-i\mathbf{k}' \cdot \mathbf{R}_j} \mathbf{u}_{\mathbf{k}'}^{(j)}. \quad (2.3)$$

ここで、式 (2.3) の両辺に位相因子 $e^{i\mathbf{k}\cdot\mathbf{R}_i}$ をかけ、また連続 k 空間での直交性を用いると、

$$\sum_{\mathbf{R}_i} e^{i(\mathbf{k}-\mathbf{k}')\cdot\mathbf{R}_i} = N_{\Omega}\delta_{\mathbf{k},\mathbf{k}'}, \quad (2.4)$$

となる。ここで、 $\delta_{\mathbf{k},\mathbf{k}'}$ はデルタ関数であり、 $k = k'$ の時 1、その他は 0 となる。式 (2.3)、式 (2.4) より、次式を得る。

$$\left(\sum_j K^{(ij)} - M_i\omega^2(\mathbf{k})I \right) \mathbf{u}_{\mathbf{k}}^{(i)} - \sum_j K^{(ij)} e^{-i\mathbf{k}\cdot\Delta\mathbf{R}_{ij}} \mathbf{u}_{\mathbf{k}}^{(j)} = 0, \quad (i = 1, \dots, N), \quad (2.5)$$

ここで、式 (2.5) において、 I は単位行列である。 $\Delta\mathbf{R}_{ij} = \mathbf{R}_j - \mathbf{R}_i$ は、 i 番目の原子に対する j 番目の原子のベクトルである。また $K^{(ij)}$ は i 番目の原子に対する j 番目の原子の力のテンソルである。

ここで注意してもらいたいのは、 j 番目の原子が最初の単位胞内ではなく、となりの単位胞内にある時である (j' 番目の原子とする)。しかしながらその時原子の周期的構造より $j = j'$ 、 $\mathbf{u}_{\mathbf{k}}^{(j)} = \mathbf{u}_{\mathbf{k}}^{(j')}$ となり、お互いに同じ原子となり最初の単位胞内で考えることができる。よってこの時、 $K^{(ij)}$ に $K^{(ij')}$ を加え、また位相因子は $e^{i\mathbf{k}\cdot\Delta\mathbf{R}_{ij'}}$ となる。よって、 \mathbf{t} を転置行列とし単位胞内に N 個の原子がある時の固有ベクトルは $\mathbf{u}_{\mathbf{k}} \equiv \mathbf{t} (\mathbf{u}_{\mathbf{k}}^{(1)}, \mathbf{u}_{\mathbf{k}}^{(2)}, \dots, \mathbf{u}_{\mathbf{k}}^{(N)})$ で表される。

ここで、 $D(\mathbf{k})$ を $3N \times 3N$ の力学的マトリックスとすると、式 (2.5) は次式になる。

$$D(\mathbf{k})\mathbf{u}_{\mathbf{k}} = 0. \quad (2.6)$$

式 (2.6) において $\mathbf{u}_{\mathbf{k}} \neq 0$ なので、各 k 点における固有値 $\omega^2(\mathbf{k})$ は永年方程式 $-\det D(\mathbf{k}) = 0$ を解くことによりもとまる。また $D(\mathbf{k})$ を 3×3 の小行列 $D^{(ij)}(\mathbf{k})$ 、 $(i, j = 1, \dots, N)$ に分けると、式 (2.5) より $D^{(ij)}(\mathbf{k})$ は次式で表される。

$$D^{(ij)}(\mathbf{k}) = \left(\sum_{j''} K^{(ij'')} - M_i\omega^2(\mathbf{k})I \right) \delta_{ij} - \sum_{j'} K^{(ij')} e^{i\mathbf{k}\cdot\Delta\mathbf{R}_{ij'}}, \quad (2.7)$$

ここで、 j'' は i 番目の原子と近接するすべての原子をさし、 δ_{ij} は $i = j$ の時 1、 $i \neq j$ の時 0 となるクロネッカーのデルタである。 $K^{(ij'')}$ はその力のマトリックスをさす。また、 j' は j 番目の原子と同じ原子をすべて表している。また δ_{ij} はこの様に周期的構造において力学的マトリックスの要素は、力のテンソル $K^{(ij)}$ と、位相因子 $e^{i\mathbf{k}\cdot\Delta\mathbf{R}_{ij}}$ になっている。

次にこの力学的マトリックスを用いて、2次元グラファイト、カーボンナノチューブのフォノン分散関係を求める。

2.2 2次元グラファイトのフォノン分散関係

この節では2次元グラファイトのフォノン分散関係の計算方法について説明する。2次元グラファイトでは、図 2.1 (a) の様にユニットセル(菱型)内に炭素原子が A 原子と B 原子の 2 個存在する。また、図 2.1(b) はグラファイトの逆格子空間であり、影をつけた部分がブリルアンゾーンとなっている。Γ、M、K 点はブリルアンゾーンにおいて対称性の高い点である。

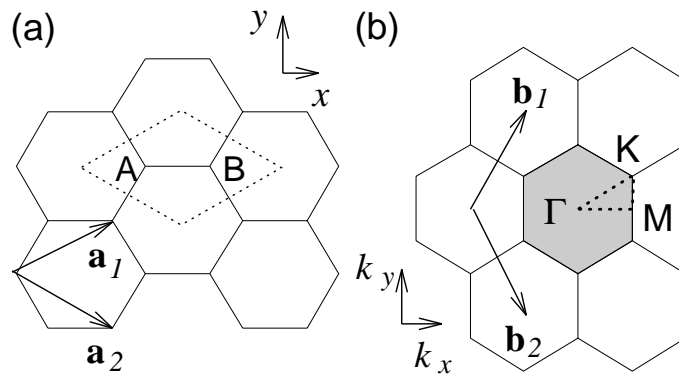


図 2.1.2D グラファイトの (a) ユニットセル(菱型) と (b) その逆格子空間(黒色)

2次元グラファイトでは、式 (2.6) の力学的マトリクス \mathcal{D} は A、B 原子各 3 つの自由度を考えて 6×6 になり、 3×3 の 4 つ小行列 D^{AA} , D^{AB} , D^{BA} , D^{BB} に分けることができ、

$$\mathcal{D} = \begin{pmatrix} D^{AA} & D^{AB} \\ D^{BA} & D^{BB} \end{pmatrix}, \quad (2.8)$$

となる。例えば、今 A 原子について第四近接原子までを考えると。図 2.2(a) を見て A 原子(黒丸)の第 1 近接原子である白丸で表した B1, B2, B3 までの 3 個の原子の他に、第 3 近接原子(白四角形)3 個、第 4 近接原子(白六角形)6 個、計 12 個までの和をとったものが D^{AB} となる(式 (2.7) の第 2 項のみ)。

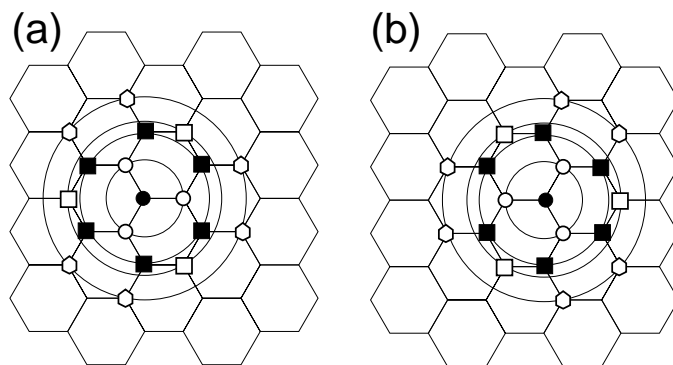


図 2.2.2D グラファイトの近接原子

次に D^{AA} であるが A 原子 (黒丸) の第 2 近接原子 A1 から A6(黒四角形) までの 6 個の原子の和 (式 (2.7) の第 2 項) の他に、式 (2.7) の第 1 項が示すように A 原子に近接する第 4 近接までのすべての原子の 18 個の各力のテンソルの和が加わる。 $D^{BB}D^{BA}$ も同じようにもとまる。

次に力のテンソルのたて方を説明する。

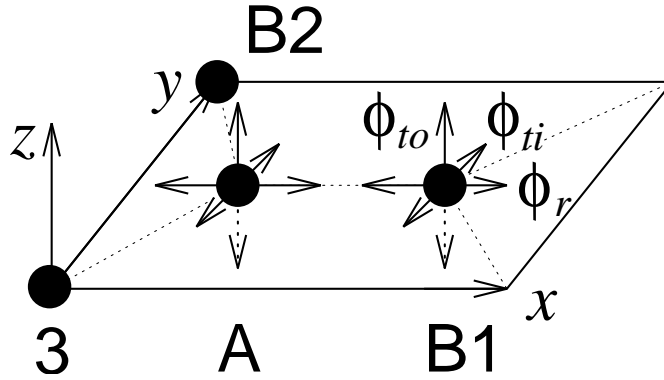


図 2.3. 炭素原子間の振動方向

今、図 2.3 の様に xyz 座標を決め A 原子と最近接原子である x 軸上の B1 原子を考えると、力のテンソルは次式によって表される。

$$K^{(A,B1)} = \begin{pmatrix} \phi_r^{(1)} & 0 & 0 \\ 0 & \phi_{ti}^{(1)} & 0 \\ 0 & 0 & \phi_{to}^{(1)} \end{pmatrix}, \quad (2.9)$$

ここで、 $\phi_r^{(n)}$ 、 $\phi_{ti}^{(n)}$ 、 $\phi_{to}^{(n)}$ はそれぞれ、第 n 近接におけるボンド方向 (radial bond-stretching)、ボンド方向と垂直でグラファイト平面内 (in-plane tangential bond-bending)、ボンド方向と垂直で平面外 (out-of-plane tangential bond-bending) の力の定数である。 B1 原子の座標は、グラファイトの基本並進ベクトルの大きさを a とおき、A 原子を原点におくと $(a/\sqrt{3}, 0, 0)$ となる。よって式 (2.7) における A と B1 原子の位相因子は、 $e^{i\mathbf{k}\cdot\Delta\mathbf{R}_{ij}} = \exp(-ik_x a/\sqrt{3})$ となる。

次に残りの A 原子の第 1 近接原子である 2 つの B2、B3 原子の力のテンソルであるが、式 (2.9) を z 軸の回りの回転テンソルを用いて、ユニタリー変換 (基底の空間の一次変換) することによってもとまる。

$$K^{(A,Bm)} = U_m^{-1} K^{(A,B1)} U_m, \quad (m = 2, 3) \quad (2.10)$$

ここで U_m はユニタリー行列で

$$U_m = \begin{pmatrix} \cos \theta_m & \sin \theta_m & 0 \\ -\sin \theta_m & \cos \theta_m & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.11)$$

の様に与えられる。よって座標が $[-a/(2\sqrt{3}), a/2, 0]$ である B2 原子における力のテンソルであるが、 $\theta_2 = 2\pi/3$ であることより U_2 がもとまる。 U_2 を式 (2.10) に代入すると A と B2 原子間の力のテンソルがもとまり、次式の様になる。

$$K^{(A,B2)} = \frac{1}{4} \begin{pmatrix} \phi_r^{(1)} + 3\phi_{ti}^{(1)} & \sqrt{3}(\phi_{ti}^{(1)} - \phi_r^{(1)}) & 0 \\ \sqrt{3}(\phi_{ti}^{(1)} - \phi_r^{(1)}) & 3\phi_r^{(1)} + \phi_{ti}^{(1)} & 0 \\ 0 & 0 & \phi_{to}^{(1)} \end{pmatrix}, \quad (2.12)$$

また、位相因子は $\exp[-ik_x a/(2\sqrt{3}) + ik_y a/2]$ となる。B3 原子も同様に、 $\theta_3 = 4\pi/3$ 、位相因子は $\exp[-ik_x a/(2\sqrt{3}) - ik_y a/2]$ であることを用いて求まる。

また、第 n 近接原子 ($n=2, 3, 4$) であるが、 x 軸上に仮想原子をおき、回転テンソルを使うことによって力のテンソルを求めることができる。力の定数のパラメータとして表 2-1 を用いる。

Radial		Tangential			
$\phi_r^{(1)} =$	36.50	$\phi_{ti}^{(1)} =$	24.50	$\phi_{to}^{(1)} =$	9.82
$\phi_r^{(2)} =$	8.80	$\phi_{ti}^{(2)} =$	-3.23	$\phi_{to}^{(2)} =$	-0.40
$\phi_r^{(3)} =$	3.00	$\phi_{ti}^{(3)} =$	-5.25	$\phi_{to}^{(3)} =$	0.15
$\phi_r^{(4)} =$	-1.92	$\phi_{ti}^{(4)} =$	2.29	$\phi_{to}^{(4)} =$	-0.58

表 2-1. 力の定数パラメーター

ここで、表 2-1 の単位は、 10^4 dyn/cm であり、第 1 近接から第 4 近接までの力の定数のパラメータを示している。

2.3 チューブのフォノン分散関係

2.3.1 zone-folding の近似の計算方法

カーボンナノチューブフォノン分散関係を求める方法の一つとして、R. A. Jshi による、二次元ブリリアンゾーンのグラファイト層を丸めて一次元ブリリアンゾーンをもつカーボンナノチューブの分散関係を求める zone-folding の近似 [1] があげられる。ここで、zone-folding の近似式 [1] を次式に示す。

$$\omega_{1D}^{m\mu}(k) = \omega_{2D}^m(k \frac{\mathbf{b}_2}{|\mathbf{b}_1|} + \mu \mathbf{b}_1), \left(\begin{array}{l} m = 1, \dots, 6, \\ \mu = 0, \dots, N-1, \end{array} \text{ and } -\frac{\pi}{|\mathbf{T}|} < k \leq \frac{\pi}{|\mathbf{T}|} \right) \quad (2.13)$$

ここで、 ω_{2D}^m は 2 次元グラファイトシートのフォノン分散関係で $\omega_{1D}^{m\mu}$ はそれに対応する 1 次元チューブの分散関係である。T は章 1.3.3 において式 (1.4) に示したチューブの並進ベクトル、 $\mathbf{b}_2, \mathbf{b}_1$ は、それぞれ並進ベクトル T と式 (1.1) のチューブの円周方向のベクトルとなるカイラルベクトル C_h の逆格子ベクトルである。また、N は式 (1.6) のユニットセル内の六員環の数で 2N 個の原子が章 1.3.2 で述べた様にユニットセル内には存在する。よってチューブのフォノンモード数は x, y, z 方向を考えて、 $3 \times 2N = 6N$ (個) となる。

しかしこの近似を使うといくつかのモードで補正が必要となってくる。

三次元空間でのチューブの音響モードとして、チューブの軸方向垂直な x, y 方向と、チューブの軸方向の z 方向のモードを一般的に考える。しかし、例えば、チューブの軸方向と垂直な方向の音響モード (x, y 方向) は、2 次元のグラファイトのどのモードとも一致しない。

また、2 次元グラファイトでは、out-plane のモードと in-plane のモードがお互いに、交じることがないのに対し、チューブの軸方向に垂直な音響モードは、左図の様にグラファイトの out-plane と in-plane のモードを交じることによってできる。そこで、上で述べた様な違い避け、より現実的な結果を得るために、次の章 2.3.2 で示す様なチューブの座標を直接使い、3 次元の力のテンソルを定義することによって、フォノン分散関係を求める方法を考えた。

2.3.2 3 次元の力のテンソル

カーボンナノチューブのフォノン分散関係を求めるために、直接表 2-1 の力の定数パラメーターを直に使い、ナノチューブの座標より 3 次元の力のテンソルを求めを使い、フォノン分散関係を求めた。ナノチューブの単位胞内に 2N 個の炭素原子があ

る時、フォノン分散関係を解くためには、 $6N \times 6N$ の力学的マトリックスを解けば良い。ここでナノチューブには、 A 、 B の 2 種類の同等な炭素原子しが存在する。よって幾何学的に同等な 2 種類の炭素原子を、今、 A_i 、 $B_j (i, j = 1 \cdots N)$ とすると、この 2 種類の原子は各、章 1.3.4 の対称ベクトル R を用いて $A1$ 、 $B1$ 原子を $p-1$ 回作用させることによって求めることができる。

$$A1 \xrightarrow{R^{p-1}} Ap, \quad \text{and} \quad B1 \xrightarrow{R^{p-1}} Bp, \quad (p = 1, \cdots, N). \quad (2.14)$$

今、 $6N \times 6N$ の力学的マトリックスを A_i 、 B_j 原子の組合せの 3×3 の $\mathcal{D}^{(A_i B_j)}$ の様な小マトリックスに分けることができる。ここで、小マトリックスとして $\mathcal{D}^{(A_i A_j)} \mathcal{D}^{(A_i B_j)} \mathcal{D}^{(B_i A_j)} \mathcal{D}^{(B_i B_j)}$, $(i, j = 1, \cdots, N)$ からなる $(2N)^2 = 4N^2$ 個のものが考えられる。ここで今 Ap 原子について考えると、 Ap について第四近接原子迄のペア ($Ap Bq$), ($Ap Aq$), ($Bp Aq$), ($Bp Bq$), ($q = 1, \cdots, 4$) の中の一つの力学的マトリックスである $\mathcal{D}^{(Ap Bq)}$ を考えると、 $K^{(Ap Bq)}$ の力の定数のテンソルは次式の様になる。

$$K^{(Ap Bq)} = (U^{-1})^{p-1} K^{(A1 Bq-p+1)} U^{p-1}, \quad (2.15)$$

U^{p-1} は、ナノチューブの軸の回りに $\Psi = 2\pi/N$ の回転マトリックスであり、次のようになる。

$$U^{p-1} = \begin{pmatrix} \cos(p-1)\Psi & \sin(p-1)\Psi & 0 \\ -\sin(p-1)\Psi & \cos(p-1)\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.16)$$

ここで、式 (2.16) は、 z 軸をチューブの軸としている。例として $K^{(A1 B1)}$ について考えるとする。

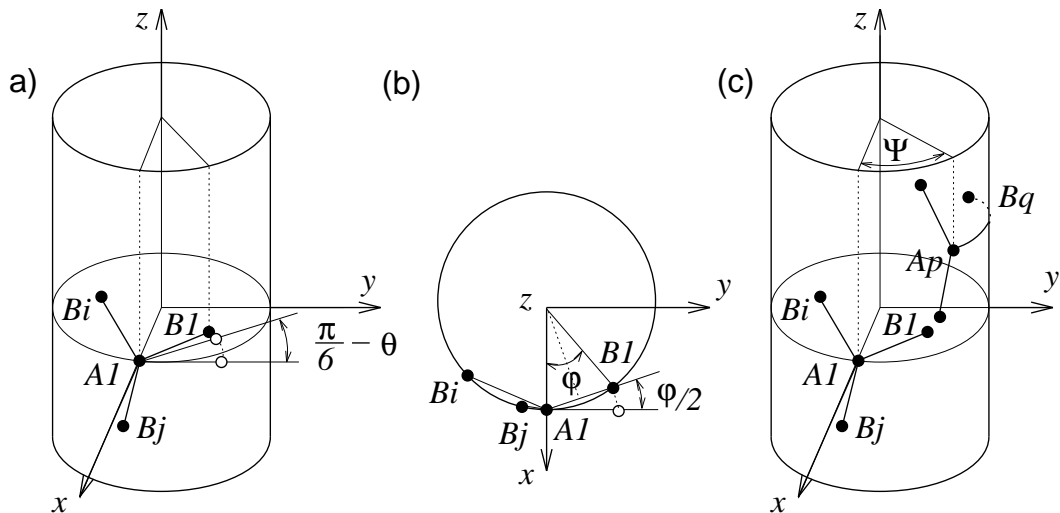


図 2.4. チューブの力のテンソルのたて方

今、 $A1$ 原子が、 x 軸上にあるとするとまず最初に、図 2.4(a) の様に、 $B1$ 原子を丸めてナノチューブにする前の 2D グラファイト上で、 x 軸の回りに回転マトリックスを用いて回転する。ここで、注意しなければならないのは、章 2.2 の 2D グラファイトの場合と x 、 z 軸が入れ替わっているので、 ϕ_{to} 、 ϕ_r を入れ変えて次式のようになる。

$$K^{(A,B1'')} = \begin{pmatrix} \phi_{to}^{(1)} & 0 & 0 \\ 0 & \phi_{ti}^{(1)} & 0 \\ 0 & 0 & \phi_r^{(1)} \end{pmatrix}, \quad (2.17)$$

$$K^{(A1B1')} = (U'')^{-1} K^{(A1B1'')} U'', \quad (2.18)$$

ここで、 U'' は x 軸の回りの回転マトリックスで、

$$U'' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_m & \sin \theta_m \\ 0 & -\sin \theta_m & \cos \theta_m \end{pmatrix}, \quad (2.19)$$

となる。ここで、 $\theta_m = (\pi/6) - \theta$ で、 θ は、章 1.3.2 で説明したカイラル角である次に、 z 軸の回りに式 (2.18) で得られた $K^{(A1B1')}$ を図 2.4(b) を見て炭素原子が、チューブの表面上にくるように $\varphi/2$ だけ回転 (U') した力のテンソル $K^{(A1B1)}$ を求める。

$$K^{(A1B1)} = (U')^{-1} K^{(A1B1')} U', \quad (2.20)$$

ここで φ は xy 平面上での $A1$ 、 $B1$ 原子間のナノチューブの軸 (z 軸) の回りの回転角である。このように $K^{(A1B1)}$ を求めることができる。

最後に、図 2.4(c) を見て Ψ は、 $A1$ 原子と A_p 原子とのナノチューブの軸の回りの回転角である。 A_p 原子においては、図 2.4(a)、(b) の操作をした後、力のテンソルをナノチューブの軸の回りに $\Psi = 2(p-1)\pi/N$ だけ回転させれば良い。

このように得られた力のテンソルに、位相因子である $\exp ik\Delta z_{ij}$ を掛けることにより、力学的マトリックスを求めることができる。ここで Δz_{ij} は、 ΔR_{ij} のチューブの波数方向 k が、ナノチューブの軸方向であることより、 z (ナノチューブの軸方向) 成分のみで良い。

2.3.3 1D ナノチューブの円筒面効果のための力の定数のパラメーターの補正

3次元カーボンナノチューブの分散関係は、章 2.3.2 で用いた方法によって求めることができる。しかしながら、表 2-1 の力の定数のパラメーターは、平面グラファイト

のパラメーターであり、ナノチューブの円筒面効果においては良く定義されたパラメーターではない。

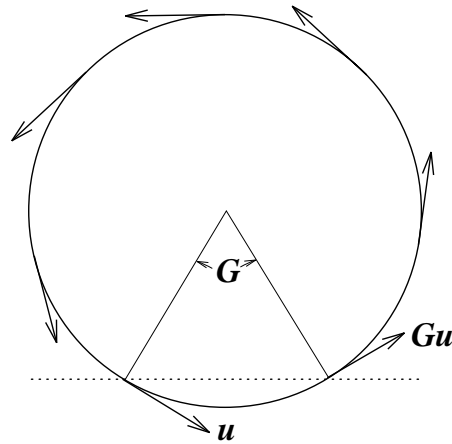


図 2.5. ナノチューブの円筒面効果

例えば、図.2.5においてナノチューブの回転モードの点における各原子の振動方向(例えば u 、 Gu 方向)は、ナノチューブの軸方向に垂直でかつ、表面に平行であることが必要である。そして、この回転モードの周波数 ω_{rot} は点において、 $\omega_{rot} = 0$ であることが、物理的に必要とされる。しかしながら、表 2-1 の力の定数のパラメーターを用いて章 2.3.2 で用いた方法を使った場合、例えば螺旋度 (10,10) のナノチューブでは、点における回転モードの周波数 $\omega_{rot}^{(10,10)}$ は、 $\omega_{rot}^{(10,10)} = 4\text{cm}^{-1}$ となってしまう。しかしながら、その他の 3 つの音響モードは点においては、 $\omega = 0\text{cm}^{-1}$ となる。今、図 2.5 において点線は、最近接原子間のボンドを表している。ここで、この 2 原子の各回転方向である u 、 Gu は、ボンド方向(図.点線)とナノチューブの軸方向で作られる平面内にはないがわかる。このことは、回転モードには out-of-plane の力の定数である ϕ_{to} の力が作用していることを示している。実際、表 2-1 のパラメーターを用いて螺旋度 (10,10) のナノチューブの各原子の点における回転モードの固有ベクトルを調べた結果、回転成分 (u 、 Gu) 方向の成分の他に、ナノチューブの表面に垂直で、半径方向にわずかながら成分(その比率は回転方向の成分に対して 10^{-5} のオーダーである)があるのがわかった。このことは、力の定数のパラメーター上にナノチューブの円筒面効果をはたらくことを示している。実際、螺旋度 (5,5) のナノチューブの場合、螺旋度 (10,10) のナノチューブよりも円筒面効果が大きいために、点における回転モードの周波数は $\omega_{rot}^{(5,5)} = 10\text{cm}^{-1}$ となる。また、逆に螺旋度 (20,20) のナノチューブの場合、螺旋度 (10,10) のナノチューブよりも円筒面効果が小さいために

めに $\omega_{rot}^{(20,20)} = 0 \text{cm}^{-1}$ となることが調べてわかった。円筒面効果は、一般的にナノチューブのフォノン分散における周波数のオーダーが 10^3cm^{-1} であるために無視できない効果である。

そこで、次に示す 2 原子間の結合長に依存するよう力の定数を補正することによって、この問題を解決した。

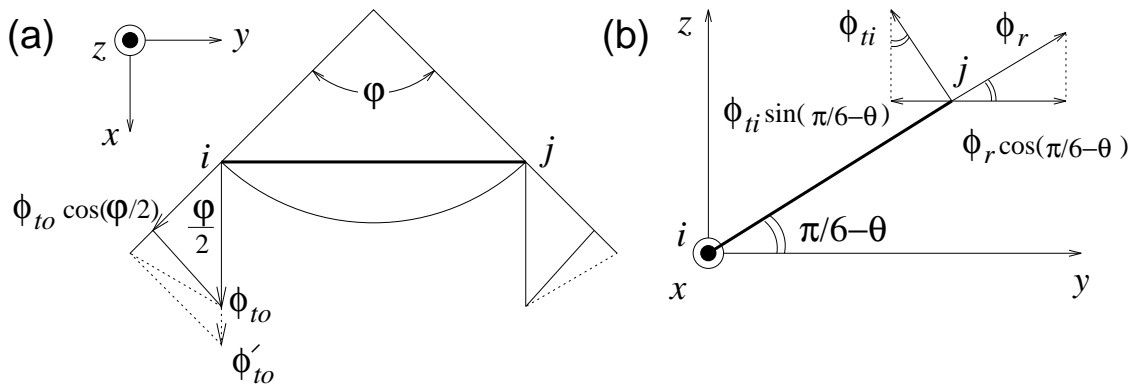


図 2.6. 力の定数パラメーターの補正 (a) チューブ円筒面、(b) グラファイト平面

まず、 ϕ_{to} (tangential out-of-plane) の成分について考える。図 2.6 (a) において例えば、 i 番目の原子における ϕ_{to} の方向は、ナノチューブの表面 (曲線) に垂直かつ、軸方向に垂直でなければいけない。しかしながら一方では、 i 番目と j 番目の原子を考えた時、 j 番目の原子は i 番目の原子をナノチューブの軸の回りに φ だけ回転した原子であり、その時の ϕ_{to} (tangential out-of-plane) の方向は、2 原子間の結合方向 (太い直線) と垂直となっており、ナノチューブの表面に垂直とはなっていない。この 2 つの方向の違いは、2 原子間の結合長に依存する角度、 $\varphi/2$ だけ違う。従って、 i, j 2 原子間の結合に垂直な ϕ_{to} (tangential out-of-plane) の成分の中で、ナノチューブの表面に垂直な動径方向の成分は、 $\phi_{to} \cos(\varphi/2)$ でになってしまう。よって、2D グラファイトを丸めてナノチューブした時の ϕ_{to} の成分の大きさが変化しないために、以下の様な補正をした。

$$\phi'_{to} = \phi_{to} + \phi_{to} \left(1 - \cos \left(\frac{\varphi}{2} \right) \right). \quad (2.21)$$

また、式 (2.21) の補正の他に同じ様な理由で、補正の 1 つとして $\phi_{to} / \cos(\varphi/2)$ が考えられるが、 $\varphi = \pi$ の時、発散してしまうので用いなかった。式 (2.21) の補正は、結合長や φ が増加するにつれて、補正が大きくなることを表している (注: 2D グラファイトを丸めてチューブにした時、結合長は短くなる)。この様に、ナノチューブの円筒表面に垂直でかつ動径方向の振動は、チューブの結合長に依存していることがわかる。

次に ϕ_r 、 ϕ_{ti} の補正であるが、同様に、 z 軸の回りに $\varphi/2$ の回転によって変化する y 軸 (ボンド方向) の力の定数のパラメーターの要素のみを考えればいいので、それぞれ、2D グラファイト上 (図 2.6(b)) ではそれぞれの要素は、 $\phi_r \cos(\pi/6 - \theta)$ 、 $\phi_{ti} \sin(\pi/6 - \theta)$ となることより次式の補正を得ることができる。

$$\phi'_r = \phi_r + \phi_r \cos\left(\frac{\pi}{6} - \theta\right) \left(1 - \cos\left(\frac{\varphi}{2}\right)\right). \quad (2.22)$$

$$\phi'_{ti} = \phi_{ti} + \phi_{ti} \sin\left(\frac{\pi}{6} - \theta\right) \left(1 - \cos\left(\frac{\varphi}{2}\right)\right), \quad (2.23)$$

ここで、 ϕ_r 、 ϕ_{ti} 、 ϕ_{to} は、表.2-1 のパラメーターである。

式 (2.21)、(2.22)、(2.23) の補正式を使って螺旋度 (10,10) のナノチューブのフォノン分散を計算した結果、 Γ 点における回転モードの周波数は、非常に小さくなる ($|\omega| < 10^{-13} \text{ cm}^{-1}$)。またその他の Γ 点における周波数の変化も ($|\omega| < 5 \text{ cm}^{-1}$ となり小さく、非常に良い近似であることがわかる。

ラマン活性モードの中でこの近似を使い大きく変化するのは最も低い周波数域にある E_{2g} モード (章 3.2.1 の図 3.11(a)) であり、 22cm^{-1} から 17cm^{-1} に変化する。この E_{2g} モードは、章 3.2.1 の図 3.12 でも説明するが、このことより、ナノチューブの円曲表面即ち、半径に敏感な周波数を持つモードであることがわかる。また、図 1.5 において低周波数域において最も強い強度を持つ A_{1g} (図 3.11(c)) モードの周波数であるが、図 3.11(c) のモードは円筒面と垂直なモードであるために 165cm^{-1} から変化しない。

2.4 カーボンナノチューブのラマン強度

2.4.1 ラマン散乱の原理

この節ではラマン強度を求める式について説明する。図 2.7 のように、分子に光を当てるとする。

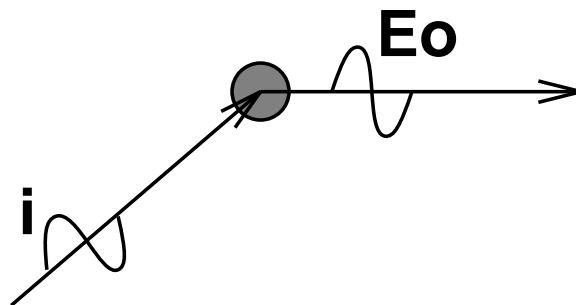


図 2.7: ラマン散乱模式図

光は電磁波であるから、入射光の電場を E_i 、その単位ベクトル e_i を振動数を ω_i と置くと、電場は式 (2.24) のように書ける。

$$\mathbf{E}_i = E_{i0} \mathbf{e}_i \cos 2\pi\omega_i t \quad (2.24)$$

分子に電場がかかると分子の電荷分布に僅かな変化が起き、双極子モーメント \mathbf{P} が誘起される。この現象を分極と呼ぶ。電場が十分に弱いときには、誘起双極子モーメント \mathbf{P} は電場に比例するので、 \mathbf{P} は式 (2.25) のように書け、 α を分極率テンソルと呼ぶ。

$$\mathbf{P} = \alpha \mathbf{E}_i \quad (2.25)$$

分子は通常、振動しており、その振動数を ω_r とすると、 α も振動数 ω_r で周期的に変化する成分を持ち、式 (2.26) のように書ける。

$$\alpha = \alpha_0 + \alpha_1 \cos 2\pi\omega_r t \quad (2.26)$$

式 (2.24) と式 (2.26) を、式 (2.25) に代入すれば入射電磁波によって誘起される双極子モーメント \mathbf{P} が求まり、式 (2.27) のようになる。

$$\begin{aligned} \mathbf{P} = & E_{i0} \alpha_0 \mathbf{e}_i \cos 2\pi\omega_i t \\ & + \frac{1}{2} E_{i0} \alpha_1 \mathbf{e}_i \cos 2\pi(\omega_i - \omega_r) t \\ & + \frac{1}{2} E_{i0} \alpha_1 \mathbf{e}_i \cos 2\pi(\omega_i + \omega_r) t \end{aligned} \quad (2.27)$$

式 (2.27) を見ると、振動数 ω_i で周期的に変化する成分の他に、振動数 $\omega_i - \omega_r$ や $\omega_i + \omega_r$ で周期的に変化する成分があることが分かる。周期的に変化する成分を持つ双極子モーメントはその振動数と同じ振動数の電磁波を放射する。したがって、入射電磁波によって誘起される双極子モーメント \mathbf{P} によって、振動数 ω_i 、 $\omega_i - \omega_r$ 、 $\omega_i + \omega_r$ を持つ電磁波が放射される。すなわち、式 (2.27) 第 1 項がレイリー散乱、第 2 項がラマン散乱 (ストークス)、第 3 項がラマン散乱 (アンチストークス) に相当する。

以上のように、振動によって分極率の変化が起きることでラマン散乱が起きる。したがって、分極率を求めればラマン強度の計算を行なうことができる。ここではこの経験的な方法として結合分極近似を用いる。

2.4.2 結合分極近似

単位胞内に \mathcal{N} 個の原子がある時結合分極近似は次式で表される。

$$I_{\eta'\eta}(\omega) \propto \omega_L \omega_S^3 \sum_{f=1}^{3\mathcal{N}} \frac{\langle n(\omega_f) \rangle + 1}{\omega_f} \left| \sum_{\alpha\beta} \eta'_\alpha \eta_\beta P_{\alpha\beta,f} \right|^2 \delta(\omega - \omega_f). \quad (2.28)$$

ここで、 ω_L 、 ω_s はそれぞれ、入射光、散乱光の光の周波数である。また、 η 、 η' は、入射光、散乱光のそれぞれ、単位分極ベクトルである。 $\omega \equiv \omega_L - \omega_s$ は、ラマンシフトである。 ω_f は、 f 番目のフォノンモードの周波数であり、

$\langle n(\omega_f) \rangle = 1/(\exp(\hbar\omega_f/k_B T) - 1)$ は、温度 $T = (k_B\beta)^{-1}$ で、 f 番目のフォノンモードの占有率を示している。 $P_{\alpha\beta,f}$ は、 f 番目のモードの分極テンソルであり、 $\alpha, \beta = x, y, z$ である。分極テンソル $P_{\alpha\beta,f}$ は、次式で与えられる。

$$P_{\alpha\beta,f} = \sum_{\ell\gamma} \left[\frac{\partial P_{\alpha\beta}}{\partial u_\gamma(\ell)} \right]_0 \chi_\gamma(\ell|f), \quad (\gamma = x, y, z, \ell = 1, \dots, \mathcal{N}, f = 1, \dots, 3\mathcal{N}) \quad (2.29)$$

ここで、 $P_{\alpha\beta}$ は、 ℓ 番目の原子の γ 座標 ($u_\gamma(\ell)$) に関する分極を表している。また、 $\chi_\gamma(\ell|f)$ は、 f 番目のモードにおける ℓ 番目の原子の固有ベクトルを示す。

式 (2.29) を計算するために、ゼロオーダー近似を使う。この近似は、結合分極パラメータを $\alpha_{\parallel} \equiv \alpha_{\parallel}(R)$ 、 $\alpha_{\perp} \equiv \alpha_{\perp}(R)$ の様な結合長 R の関数とし、結合と寄与しない原子 (第一近接原子のみ) の振動は無視できる近似である。よって、この近似に従うと次式を得ることができる。

$$P_{\alpha\beta} = \frac{1}{2} \left[\sum_{\ell,B} \left\{ \frac{\alpha_{\parallel}(B) + 2\alpha_{\perp}(B)}{3} \right\} \delta_{\alpha\beta} + \left\{ \alpha_{\parallel}(B) - \alpha_{\perp}(B) \right\} \left(\frac{R_\alpha(\ell, B)R_\beta(\ell, B)}{R(\ell, B)^2} - \frac{1}{3}\delta_{\alpha\beta} \right) \right], \quad (2.30)$$

ここで、 B は単位胞内において ℓ 番目の原子と結びついているボンドを示し、 $\mathbf{R}(\ell, B)$ は、 ℓ 番目の原子から、ボンド B によって、結合している ℓ' 番目の原子へのベクトルを示す。 $R_\alpha(\ell, B)$ 、 $R(\ell, B)$ はそれぞれ、 $\mathbf{R}(\ell, B)$ の α 成分の要素、 $\mathbf{R}(\ell, B)$ の大きさである。また、 $\alpha_{\parallel}(B)$ 、 $\alpha_{\perp}(B)$ は、ボンド B に関してそれぞれ平行、垂直方向の分極率である。ここで、先ほど述べたように、 $\alpha_{\parallel}(B)$ 、 $\alpha_{\perp}(B)$ は、結合長 $R(\ell, B)$ の関数とする。

$$\mathbf{R}(\ell, B) = \mathbf{R}_0(\ell, B) + \mathbf{u}(\ell') - \mathbf{u}(\ell), \quad (2.31)$$

式 (2.29) の u_γ に関するところは、 $R(\ell, B)$ を用いて次式の様に変形できる。

$$\frac{\partial}{\partial u_\gamma(\ell)} = \sum_B \frac{\partial}{\partial R(\ell, B)} \frac{\partial R(\ell, B)}{\partial u_\gamma(\ell)} = - \sum_B \frac{\partial}{\partial R(\ell, B)} \frac{R_\gamma(\ell, B)}{R(\ell, B)}. \quad (2.32)$$

また、次の関係式を使い、 ℓ 番目の原子と結び付くボンドの合計をとる。

$$\frac{\partial R_\alpha(\ell, B)}{\partial u_\gamma(\ell)} = -\delta_{\alpha\gamma}, \quad (2.33)$$

$$\frac{\partial R(\ell, B)}{\partial u_\gamma(\ell)} = \sum_\alpha \frac{\partial R(\ell, B)}{\partial R_\alpha(\ell, B)} \frac{\partial R_\alpha(\ell, B)}{\partial u_\gamma(\ell)} = -\frac{\partial R(\ell, B)}{\partial R_\gamma(\ell, B)} = -\frac{R_\gamma(\ell, B)}{R(\ell, B)}. \quad (2.34)$$

また、式 (2.30)、(2.32) より、 $\frac{\partial P_{\alpha\beta}}{\partial u_\gamma(\ell)}$ は、次ぎの $\frac{\partial \alpha}{\partial u_\gamma}$ 、 $\frac{\partial R_\alpha(\ell, B)}{\partial u_\gamma}$ 、 $\frac{\partial R(\ell, B)}{\partial u_\gamma}$ の項があることに注意して、式 (2.32)、(2.33)、(2.34) を用いて $P_{\alpha\beta,f}$ を求めることができる。

$$\begin{aligned} P_{\alpha\beta,f} = & -\sum_{\ell B} \left[\frac{\mathbf{R}_0(\ell, B) \cdot \vec{\chi}(\ell|f)}{R_0(\ell, B)} \times \left\{ \left(\frac{\alpha'_{\parallel}(B) + 2\alpha'_{\perp}(B)}{3} \right) \delta_{\alpha\beta} \right. \right. \\ & + \left. \left(\alpha'_{\parallel}(B) - \alpha'_{\perp}(B) \right) \left(\frac{R_{0\alpha}(\ell, B)R_{0\beta}(\ell, B)}{R_0(\ell, B)^2} - \frac{1}{3}\delta_{\alpha\beta} \right) \right\} \\ & + \left(\frac{\alpha_{\parallel}(B) - \alpha_{\perp}(B)}{R_0(\ell, B)} \right) \left\{ \frac{R_{0\alpha}(\ell, B)\chi_\beta(\ell|f) - R_{0\beta}(\ell, B)\chi_\alpha(\ell|f)}{R_0(\ell, B)} \right. \\ & \left. \left. - \frac{\mathbf{R}_0(\ell, B) \cdot \vec{\chi}(\ell|f)}{R_0(\ell, B)} \times \frac{2R_{0\alpha}(\ell, B)R_{0\beta}(\ell, B)}{R_0(\ell, B)^2} \right\} \right], \quad (2.35) \end{aligned}$$

ここで、

$$\alpha'_{\parallel}(B) \equiv \frac{\partial \alpha_{\parallel}(B)}{\partial R(\ell, B)}, \quad \text{と} \quad \alpha'_{\perp}(B) \equiv \frac{\partial \alpha_{\perp}(B)}{\partial R(\ell, B)}, \quad (2.36)$$

分極率パラメータの微分である。 $\alpha_{\parallel}(B)$ 、 $\alpha_{\perp}(B)$ 、 $\alpha'_{\parallel}(B)$ 、 $\alpha'_{\perp}(B)$ は、2つの炭素原子間、ハイドロカーボン原子間のボンド結合長の関数として与えられる。表.2-2に C_{60} を含むさまざまな炭素クラスターの実験データと fitting させたダブルボンド、シングルボンドの分極率パラメータをのせる。

まず最初に、カーボンナノチューブの分極率パラメータとして、ダブルボンド (1.40)、シングルボンド (1.46) の C_{60} のパラメータを使い、ナノチューブの軸がランダムにある時のラマン強度を計算してみた。しかしながら、表.2-2 をみてわかるように、各パラメーターとも近いパラメータになっているが、さまざまなグループで、ことなっているのがわかる。

しかしながら、ラマン強度を計算してみた結果、最も低い周波数をもつラマン活性モードである E_{2g} (章 3.2.1 図 (3.13)) モードを除いては、分極パラメータの小さな変化には、敏感ではなかった。また、この E_{2g} モードは異方的項に関係するパラメータである $\alpha_{\parallel} - \alpha_{\perp}$ のパラメータに非常に敏感なモードであることがわかった。今回、SWCN の分極率パラメータとして、(d) のパラメータを考え、用いた。この分極率パラメータと結合分極近似を使いラマン強度を計算した。

表 2-2. 結合長に寄与するナノチューブと関係する炭素クラスターの分極率パラメータ

Molecule	Bond Lengths [Å]	$\alpha_{\parallel} + 2\alpha_{\perp}$ [Å ³]	$\alpha_{\parallel} - \alpha_{\perp}$ [Å ³]	$\alpha'_{\parallel} + 2\alpha'_{\perp}$ [Å ²]	$\alpha'_{\parallel} - \alpha'_{\perp}$ [Å ²]
CH ₄ ^{a)}	C - H (1.09)	1.944			
C ₂ H ₆ ^{a)}	C - C (1.50)	2.016	1.28	3.13	2.31
C ₂ H ₄ ^{a)}	C = C (1.32)	4.890	1.65	6.50	2.60
C ₆₀ ^{b)}	C - C (1.46)		1.28	2.30 ± 0.01	2.30 ± 0.30
	C = C (1.40)		0.32 ± 0.09	7.55 ± 0.40	2.60 ± 0.36
C ₆₀ ^{a)}	C - C (1.46)		1.28 ± 0.20	1.28 ± 0.30	1.35 ± 0.20
	C = C (1.40)		0.00 ± 0.20	5.40 ± 0.70	4.50 ± 0.50
SWCN ^{c)}	C = C (1.42)		0.07	5.96	5.47
SWCN ^{d)}	C = C (1.42)		0.04	4.7	4.0

^{a)} D. W. Snoke and M. Cardona [14].

^{b)} S. Guha et al. [3].

^{c)} E. Richter et al. (unpublished data which is used in their work).

^{d)} 今回用いたパラメーター

第 3 章

結果・考察

この章では、章 3.1.1、章 3.1.2 ではそれぞれグラファイトとチューブのフォノン分散関と状態密度について、章 3.1.3 ではチューブの音速度の螺旋度依存性についてそれぞれ説明する。また章 3.2.1 でカーボンナノチューブのラマン活性周波数が半径に依存していることを示し、章 3.2.2 ではチューブのラマン強度の角度依存性を示す。次いで章 3.2.4 でチューブのラマン強度の端の効果について説明する。

3.1 フォノン分散関係

3.1.1 グラファイトのフォノン分散関係

図 3.1は表.2-1 の力の定数を用いた第 4 近接原子までを考慮した 2 次元グラファイトの分散関係 (右) と状態密度 (左) である。フォノン分散関係は、縦軸、横軸にそれぞれ周波数 ωcm^{-1} 、波数をとっている。また、状態密度は、縦軸、横軸にそれぞれ周波数 ωcm^{-1} 、状態密度を単位 $\text{states/C-atom/cm}^{-1}$ でとっている。状態密度を周波数に沿って積分すると 3 になる。ここで、波数 k の Γ 、M、K 点は、章 3.1.1 図 2.1(b) のグラファイトのブリルアンゾーンにおいて高い対称性をもつ点ある。グラファイトは、ユニットセル内に 2 個の原子があるので、 x 、 y 、 z 方向の各原子 3 つの自由度を考えて全部で $2 \times 3=6$ (個) のモードが現れる。その内 3 つが光学モードであり、残りの 3 つが音響モードである。

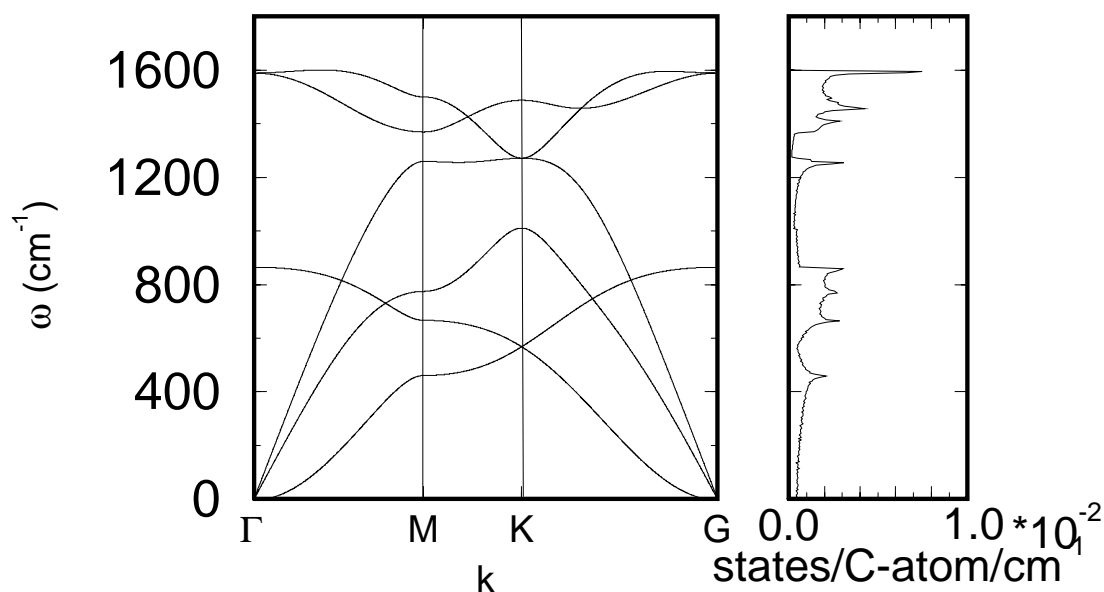


図.3.1 2D グラファイトのフォノン分散関係と状態密度

グラファイトの音響モードを 3.2 に示す。

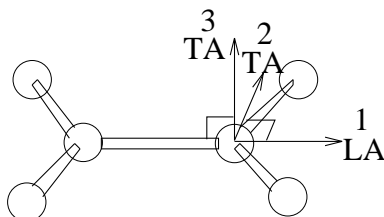


図.3.2 2D グラファイトの音響モード

音響モードであるが、図.3.2に見られる様な2つの炭素原子のボンド方向に振動する LA モード、ボンド方向と垂直で in-plane の TA モードと、ボンド方向と垂直で out-of-plane の TA モードの3つがある。それぞれの音速は 21.11km/s , 14.99km/s , 0km/s である。

ここで注目してもらいたいのは、グラファイトの音響モードの特徴である音速が k に比例する2つの in-plane のモードである LA, TA モードの他に、 k^2 に比例する out-of-plane 方向に振動する TA モードがある点である。

グラファイトの3つの光学モードを図 3.3 に示す。

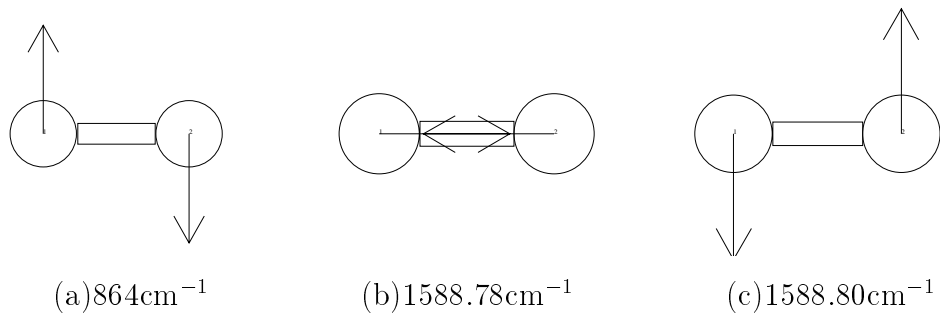


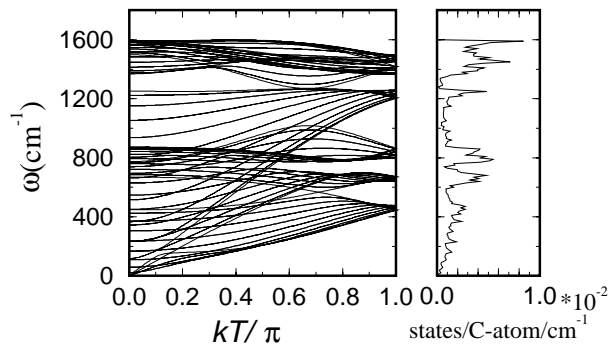
図 3.3 グルファイトの、高周波域のフォノンモード

図 3.1の右に 2D グラファイトのフォノン状態密度を横軸、縦軸にそれぞれ状態密度を $\text{state/C-atom/cm}^{-1} \times 10^{-2}$ の単位で、周波数 $\omega \text{ cm}^{-1}$ で示す。

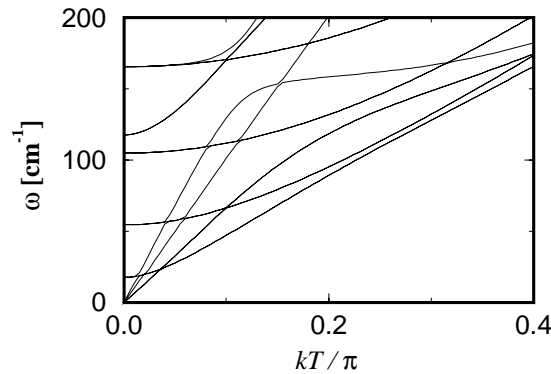
1 つの炭素原子に対して、 x, y, z の 3 つの振動方向があるので ω にそって積分すると 3 になる。ここでフォノン状態密度のピークは、 $\omega = 1600, 1400, 1200, 800, 400 \text{ cm}^{-1}$ 付近に、8 本のピークが出ているのがわかる。

3.1.2 チューブのフォノン分散関係

この節では表.2-1 の力の定数を用いた第 4 近接原子までを考慮した単層カーボンナノチューブの分散関係と状態密度を示す。図 3.4(a) に、その 1 例として螺旋度 (10,10) の armchair 型チューブのフォノン分散関係 (右) の全体図とフォノン状態密度 (左)、図 3.4 (b) は、フォノン分散関係の全体図の点付近の拡大図である。フォノン分散関係は、縦軸、横軸にそれぞれ周波数 $\omega \text{ cm}^{-1}$ 、波数をとっている。また、状態密度は、縦軸、横軸にそれぞれ周波数 $\omega \text{ cm}^{-1}$ 、状態密度を単位 $\text{states/C-atom/cm}^{-1}$ でとっている。状態密度を周波数に沿って積分すると 3 になる。



(a)



(b)

図 3.4.(a) 螺旋度 (10,10) のチューブのフォノン分散関係と状態密度
 (b)(a) のフォノン分散関係の 点付近の拡大図

ここで、チューブのユニットセル内の炭素原子を $2N$ (個) とすると、章 2.3.2 で述べたように、フォノン数は 1 炭素原子に対する x, y, z 方向を考えて、 $2N \times 3 = 6N$ (個) の分散が現れる。例えば図 3.5(a) の様な螺旋度 (10,10) のチューブの分散関係では、ユニットセル内の炭素原子数は 40(個) なので、フォノン数は $3 \times 40 = 120$ (個) の分散が現れる。そのうち、縮重度 2 が 54 個、残りが縮重度 1 で 12 個あり、計 66 個の枝をもつ。

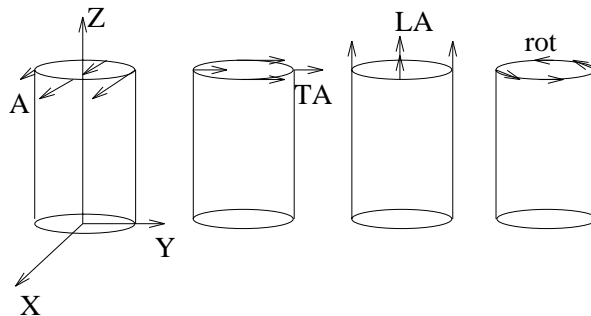


図 3.5. カーボンナノチューブの音響モード

ここで注目してもらいたいのは、グラファイトの音響モードが波数 k に比例する in-plane の LA, TA と k^2 に比例する out-of-plane の TA モードの 3 つだったのに対して、チューブの音響モードは、図 3.5 に見られる様にチューブの軸方向に垂直な x, y 方向の 2 重に縮重した TA モード、チューブの軸方向の軸方向の LA モード、またグラファイトに見られないチューブの軸の回りを回転する RA モードを合わせて 4 つの音響モードをもつ。

またグラファイトのフォノン分散関係とのもう一つの大きな違いは、グラファイトの音響モードでは章 3.1.1の図.3.1に見られる様に波数 k^2 に比例する分散が現れるのに対して、チューブの音響モードでは、チューブの波数方向は軸方向に 1 次元なので 4 つの音響モードすべて波数 k に比例する分散が現れる点である。

また、チューブのフォノン状態密度であるが高周波数域におけるフォノン分散関係は 2 つの炭素原子間の振動の影響を受けるが低周波数域帯においてはチューブの構造全体の振動の影響を受けるので、章 3.1.1の図 3.1におけるグラファイトのフォノン状態密度と図 3.4のチューブのフォノン状態密度と比べてわかる様に高周波数域 ($1200\text{cm}^{-1} \sim 1600\text{cm}^{-1}$) においてはグラファイトの状態密度のピークの位置などほぼ一致するが、低周波数域帯においては 800cm^{-1} 付近のピークの位置や、点付近のにもチューブの方は周波数に対してフォノン状態密度が線形に立ち上がるなどの違いが現れている。

ここで、チューブの音響モードの音速度であるが、LA、TA、RA モードそれぞれ、螺旋度 (10,10) だと $v_{LA}^{(10,10)} = 20.35\text{km/s}$ 、 $v_{TA}^{(10,10)} = 9.43\text{km/s}$ 、 $v_{RA}^{(10,10)} = 14.99\text{km/s}$ である。ここで、チューブの LA、RA モードは in-plane のモードで、グラファイトの章 3.1.1の図 3.2に見られる様に in-plane の LA、TA モード方向と一致しているため、グラファイトの音速度である $v_{LA}^G = 21.111\text{km/s}$ 、 $v_{RA}^G = 14.99\text{km/s}$ と非常に近い値になっていることがわかる。

次に、チューブの音速度が半径と螺旋度のどちらに依存しているか調べた結果を次の章 3.1.3に示す。

3.1.3 カーボンナノチューブの音速度の螺旋度依存性

図.3.6に半径 r が $6.467(\text{Å}) \leq r \leq 6.880(\text{Å})$ までの半径の近い ($\delta r = 0.423(\text{Å})$) 螺旋度が違うチューブのそれぞれの音響モードの音速度の螺旋度依存性を横軸にカイラル角 (章 1.3.2の式 (1.3)、縦軸に音速度をとって示す。

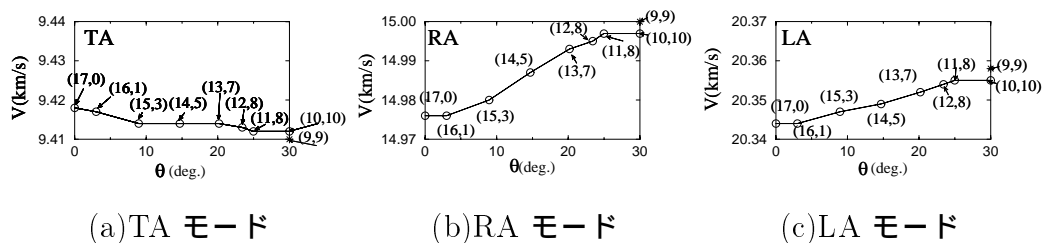


図 3.6. チューブの音響モードの螺旋度依存性

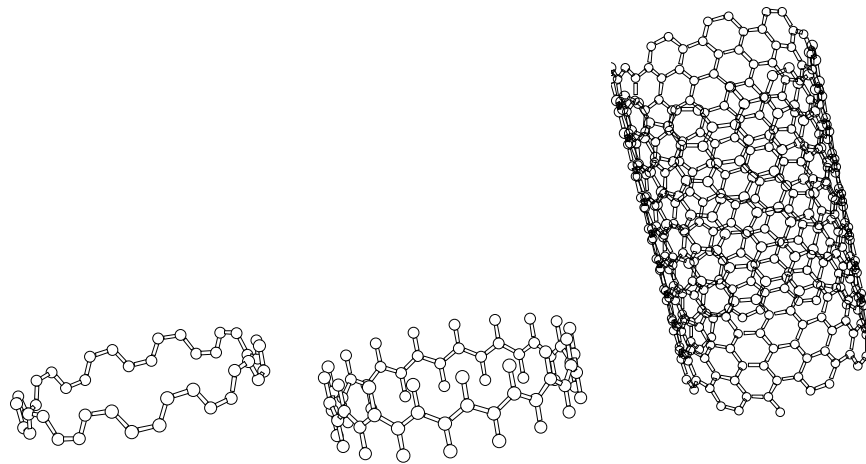
ここでチューブの半径依存性であるが、例えば (10,10) のチューブ (半径 $r^{(10,10)} = 6.780(\text{Å})$) と螺旋度の等しい (9,9) のチューブ (半径 $r^{(9,9)} = 6.102(\text{Å})$) の LA,TA,RA の音速度はそれぞれ $v_{LA}^{(9,9)} = 20.358 \text{ km/s}$ 、 $v_{TA}^{(9,9)} = 9.410 \text{ km/s}$ 、 $v_{RA}^{(9,9)} = 15.000 \text{ km/s}$ であり、 $r^{(10,10)} - r^{(9,9)} = 0.678(\text{Å})$ は $\delta r = 0.423(\text{Å})$ と比べて半径の差が約 1.6 倍であるにもかかわらず、図 3.6に見られる様に螺旋度 (9,9) の音速度 (* 印) のチューブの螺旋度 (10,10) のチューブの音速度に対する変位は他の螺旋度のチューブの螺旋度 (10,10) の音速度に対する変位に比べて小さいのでチューブの音響モードの音速度はチューブの半径よりも螺旋度に依存していることがわかる。

また図 3.6を見て、チューブの LA,TA モードの音速度はカイラル角に対し、 10^{-2} km/s の order で変位し、カイラル角が大きくなるにつれて速くなる。チューブの軸方向に垂直な TA モード (x, y 方向) の音速度の螺旋度依存性であるが、グラファイトの out-of-plane の音速度が波数 k^2 に比例する TA モードの影響を受けているので、解明するのが難しいが、チューブの RA、TA モードの螺旋度依存性とは逆にカイラル角が大きくなるにつれて音速度が遅くなる。

3.2 カーボンナノチューブのラマン強度

3.2.1 カーボンナノチューブがランダムにある場合のラマン強度

この節では、カーボンナノチューブのラマン強度について説明する。まず、カーボンナノチューブのラマン強度が半径に依存しているか、螺旋度に依存しているかを調べるために図 3.7の半径の近い螺旋度の違う 3 つのチューブ ((a) 螺旋度 (10,10) armchair 型チューブ、半径 $r^{(10,10)} = 6.78(\text{Å})$ 、(b) 螺旋度 (17,0) zigzag 型チューブ、半径 $r^{(17,0)} = 6.66(\text{Å})$ 、(c) 螺旋度 (11,8) カイラル型チューブ、半径 $r^{(11,8)} = 6.47(\text{Å})$) のラマン強度を調べる。



(a)(10,10), $r^{(10,10)} = 6.78(\text{Å})$ (b)(17,0), $r^{(17,0)} = 6.66(\text{Å})$ (c)(11,8), $r^{(11,8)} = 6.47(\text{Å})$

図 3.7. 螺旋度の違う半径の近い 3 つのチューブ

図 3.10 は、図 3.7 の半径の近い螺旋度の違う 3 つのチューブがランダムにある時、たくさんの方向からチューブに光を照射して各方向のラマン強度の和をとったものである。また、横軸にラマンシフト、縦軸にラマン強度を表す。また、ラマン強度比は VV 測定における各カーボンナノチューブの低周波領域 A_{1g} モード (ブリッジングモード) の強度を最大値としそれぞれのモードにおける強度を規格化している。ここで、ラマン強度を求めるにあたって VV、VH の二つの測定方法を考える。

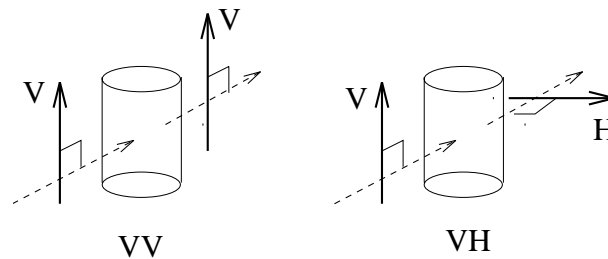


図 3.8. ラマン強度の 2 つの測定方法

VV 測定とは、図 3.8 の様に入射光、散乱光の偏光がそれぞれ平行、VH 測定とは、入射光、散乱光の偏光がそれぞれ垂直な場合の測定方法である。

まず、群論によるチューブのラマン活性モードの対称性について説明する。

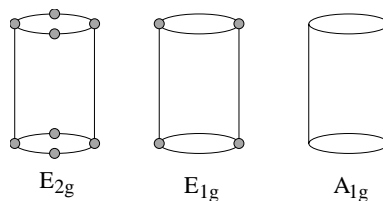


図 3.9 チューブのラマン活性モードの対称性

図 3.9 を見て簡単にラマン活性モードの対称性は、チューブの軸の回りに振動しない節が、 E_{2g} 、 E_{1g} 、 A_{1g} モードはそれぞれ 4、2、0(個) あるモードになる。このことからナノチューブのラマン強度は、螺旋度が代わっても、ラマン活性モードの対称性は変わらないので、似たようなラマン強度が得られる。

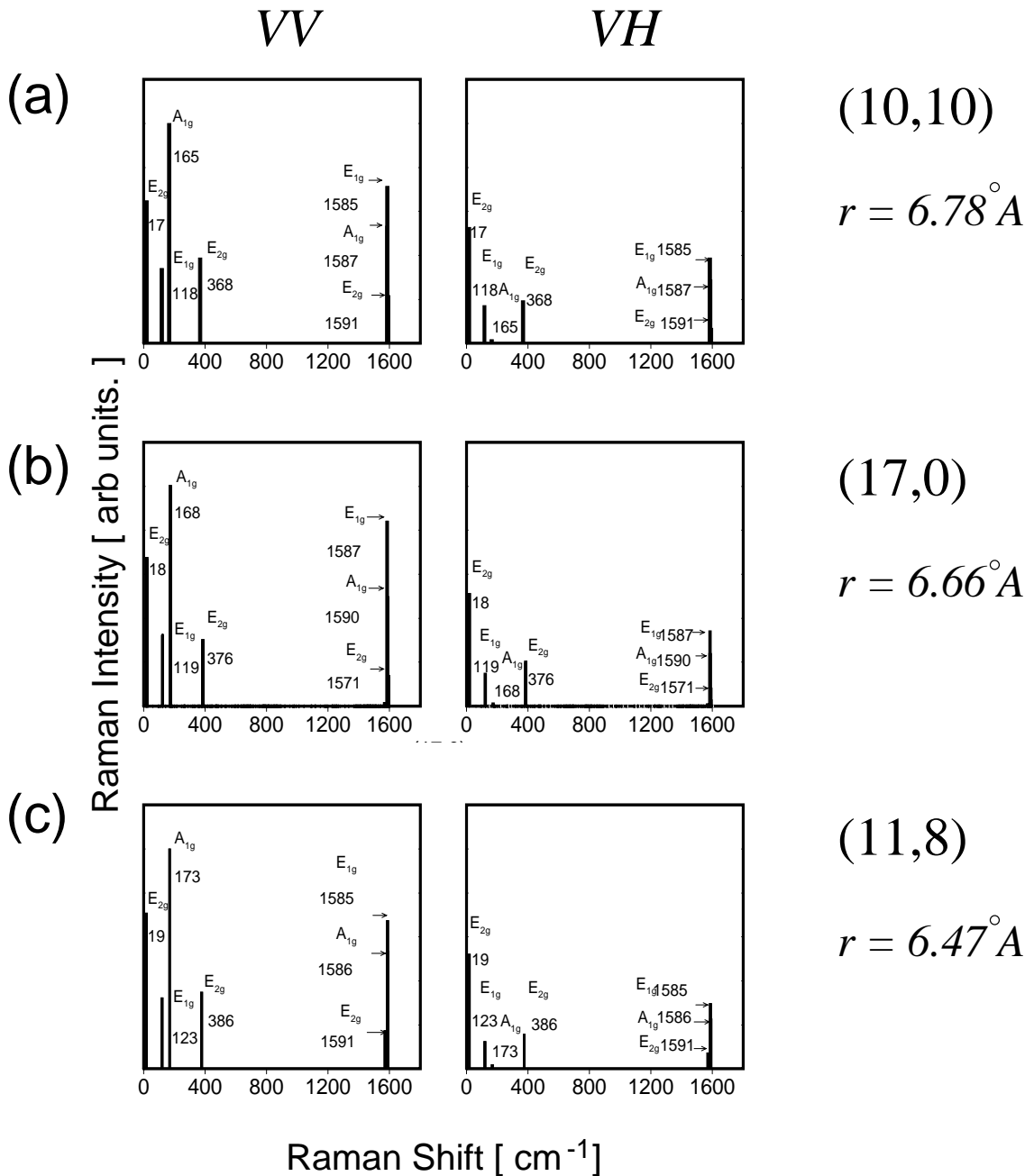


図 3.10. カーボンナノチューブがランダムにある場合のラマン強度

今、図 3.10 の (a),(b),(c) を見てわかるように VV, VH 測定ともラマン強度が強く出るラマン活性モードは、低周波数帯 ($\omega \geq 500(\text{cm})^{-1}$) における $E_{2g}, E_{1g}, A_{1g}, E_{2g}$ モードの 4 つと高周波数帯 ($\omega \leq 1500(\text{cm})^{-1}$) における E_{1g}, A_{1g}, E_{2g} モードの 3 つ、計 7 つのモードである。チューブのラマン活性モードの一例として螺旋度 (10,10) のモードを図 3.11 に示す。

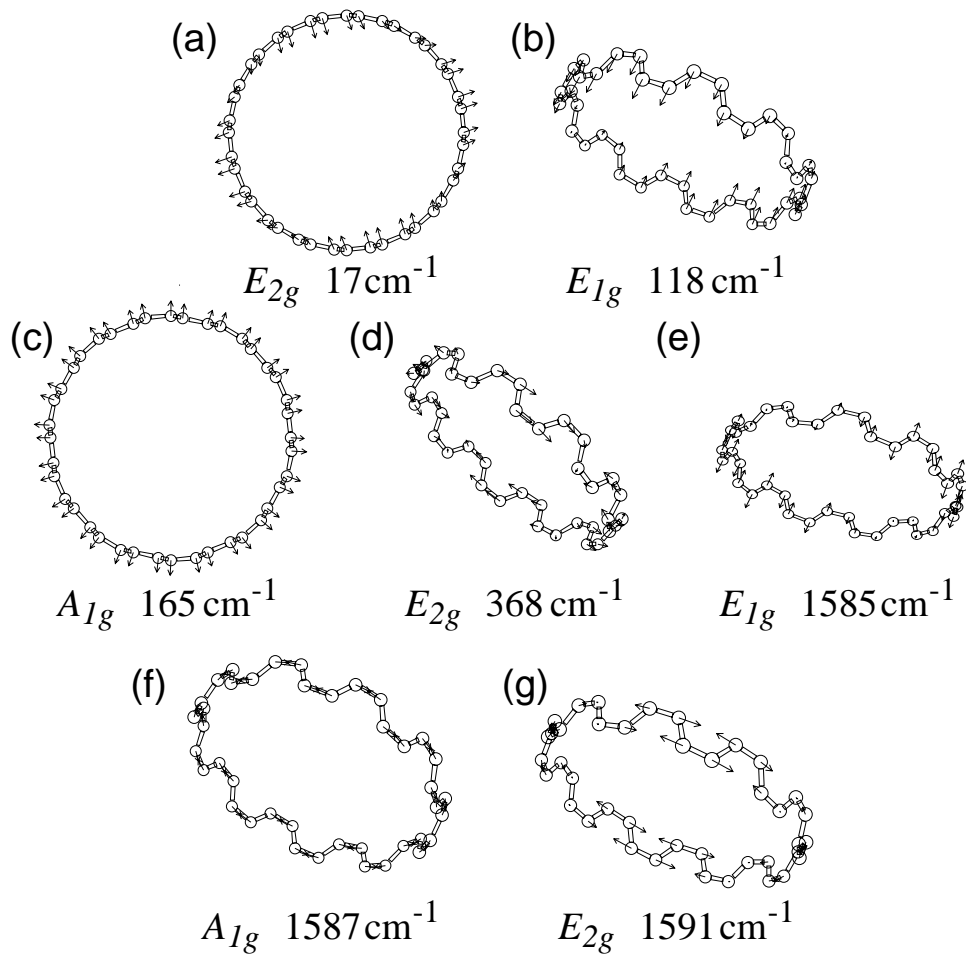


図 3.11.(10,10) ラマン活性モード

まず、チューブのラマン強度を考察するにあたって (1) 低周波数領域 ($\omega \leq 500\text{cm}^{-1}$)、(2) 中周波数領域 ($500\text{cm}^{-1} \leq \omega \leq 1500\text{cm}^{-1}$)、(3) 高周波数領域 ($\omega \geq 1500\text{cm}^{-1}$) の 3 つの領域に分けて考察する。

(1) 低周波数領域 ($\omega \leq 500\text{cm}^{-1}$) ラマン強度

低周波数領域ラマン強度において最も注目してもらいたいのは、図 3.10 を見てわかるように VV 測定においては低周波数帯 A_{1g} モードである図 3.11(c) の様な円筒形の

同径方向に広がるようなモードであるブリージングモードが 7 つのラマン活性モードの中で強度が最も強いピークが出るのに対して、 VH 測定になると強度のピークが非常に小さくなる点である。一方、低周波数領域 E_{1g} 、 E_{2g} モードでの強度は VH 測定においても VV 測定との強度の違いこそあれ表れている。図 3.10 における VV 測定と VH 測定を比べて、低周波数領域における 4 つのラマン活性モードの周波数は螺旋度にはなく半径に依存していることがわかる。そこでどのような半径依存性をしているかを次の図 3.12 に両対数で、横軸、縦軸に各カーボンナノチューブの半径 r (), ラマン活性周波数 $\omega \text{ cm}^{-1}$ をプロットして示す。

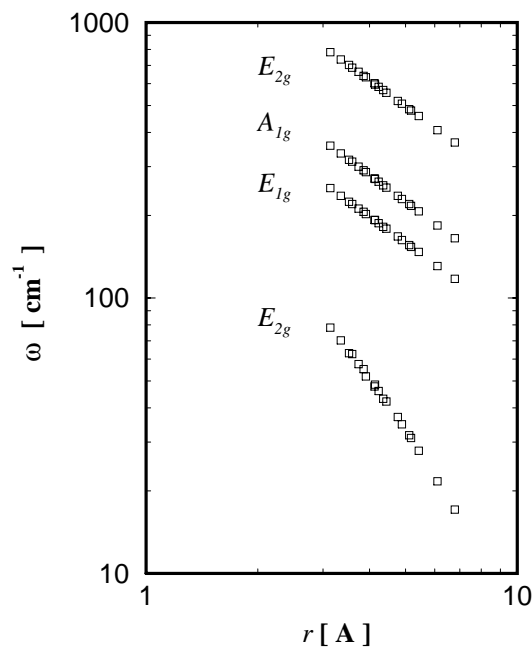


図 3.12. 低周波ラマン活性周波数の半径依存性

ここで図 3.12 は、螺旋度 (n, m) ($8 \leq n \leq 10, 0 \leq m \leq n$) のユニットセル内の原子数が 800 以下の 19 個のチューブの半径依存性を示している。また各ラマン活性モードを次の図 3.13 に示す。

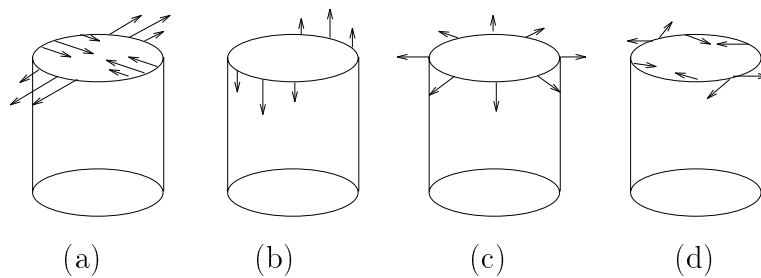


図 3.13. 低周波ラマン活性モード

ここで注目してもらいたいのは E_{2g} 、 A_{1g} 、 E_{1g} (それぞれ、図 3.13 の (d)、(c)、(b)) モードの活性周波数 ω はチューブの半径 r^{-1} に比例しているのに対して、最も低い周波数にあるラマン活性モードである円筒形のチューブが楕円形に変形する様なモードである E_{2g} (図 3.13 の (a)) モードは半径 r^{-2} に比例して、半径の変化に非常に敏感なモードである点である。ここで、この E_{2g} モードにおけるラマン強度は実験では、非常に低い周波数領域にあるのと同時に、 0cm^{-1} 付近にレイリー散乱があるために観測されていないが、図 3.12 を見てわかるように、例えば螺旋度 (8,0)、半径 $r = 3.1315(\text{Å})$ の様なナノチューブの場合、 E_{2g} モードの活性周波数 ω は 10cm^{-1} 付近になる為にラマン強度が実験でも観測されるはずである。

またラマン強度の実験 [2] [11] において最も強く観測されるブリージングモードである A_{1g} モード (3.13) のラマン活性周波数であるが、図 3.12 の ω - r 関係より、次式が得られる。

$$(r) = (10,10) \left\{ \frac{r_{(10,10)}}{r} \right\}^{-1.10017 \pm 0.0007} \quad (3.1)$$

ここで、 $r_{(10,10)}$ 、 $\omega_{(10,10)}$ はそれぞれ、螺旋度 (10,10) のナノチューブの半径 $r = 6.785(\text{Å})$ 、低周波 A_{1g} モードのラマン活性周波数 $\omega = 165\text{cm}^{-1}$ である。また図 3.12 において E_{1g} モード (図 3.13 の (b)) と A_{1g} (図 3.13 の (c)) モードの周波数は近い周波数領域に表れているが、 E_{1g} モードにおける強度は A_{1g} モードにおける強度よりはるかに実験ではラマン強度は弱いので、式 (3.1) は実験で、低周波 A_{1g} モードのラマン強度を観測するのにおおいに役に立つはずである。図 3.14 に式 (3.1) の曲線と低周波 A_{1g} モードのラマン活性周波数の半径依存性の実験値 (○ 印)[11] を横軸に周波数 ω 、縦軸に半径 r をプロットして示す。

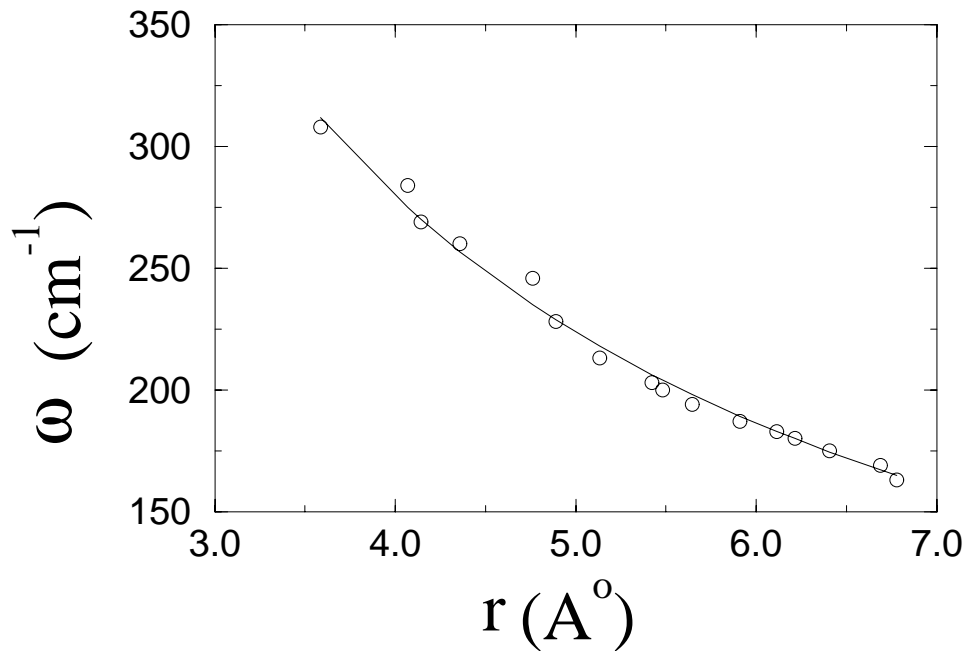


図.3.14 A_{1g} モード (ブリージングモード) のラマン活性周波数の半径依存性の実験値 (○印)[11] と理論値 (曲線)

図 3.14の式 (3.1) の理論値と実験値 [11] を比べてわかるように、実験値は理論値の曲線の非常に近い範囲に表れて式 (3.1) は非常に妥当な式であることがわかる。次に誤差について述べるが、誤差が大きいのが 2 点あり螺旋度 (6,6)、半径 $r = 4.068$ () のチューブと螺旋度 (8,6)、半径 $r = 4.7621$ () のチューブでそれぞれ実験値、理論値は (6,6) の場合 284, 275 cm^{-1} 、(8,6) の場合 246, 235 cm^{-1} である。その他のチューブは 4 cm^{-1} 以下である。よって実験値との誤差は、 $\pm 11\text{cm}^{-1}$ で小さい。

(2) 中周波数領域 ($500\text{cm}^{-1} \leq \omega \leq 1500\text{cm}^{-1}$) ラマン強度

章 1.6の図 1.5の螺旋度 (10,10) 単層カーボンナノチューブのラマン強度の実験 [2] では、中周波数領域にも弱い強度の強度がいくつか観測されている。しかしながら、今回、端の無い固体のナノチューブにおいてラマン強度を計算した図.3.10の結果においては、螺旋度 (10,10) のナノチューブのラマン強度において中周波数領域における強度は表れなかった。この計算からではなぜ、この周波数領域にこれらの強度が表れるか説明できない。

今回、中周波数領域におけるラマンスペクトルは、チューブの端の効果によるもの考え、ラマン強度を計算した。この計算結果は章 3.2.4に示す。

(3) 高周波数帯 ($\omega \leq 1500\text{cm}^{-1}$) ラマン強度

ここで、カーボンナノチューブは格子は A 原子と B 原子の 2 つの副格子からなっている (章 3.1.1 参照)。

高周波数帯ラマン強度において最も注目してもらいたいのは、図 3.10 を見てわかるように、例えば螺旋度 (10,10) のナノチューブのラマン強度において $\omega = 165\text{cm}^{-1}$ の低周波 A_{1g} モード (図 3.11 の (c)) におけるラマン強度は、VV 測定比べて VH 測定では、非常に小さくなるのに対して、 $\omega = 1587\text{cm}^{-1}$ 高周波 A_{1g} モードにおけるラマン強度は、強度に違いこそあれ大きな変化は見られない。これは A 原子と B 原子の 2 つの副格子の振動方向に関係している。例えば低周波 A_{1g} モード (図 3.11(c)) は、A、B の 2 つの原子がチューブの単位胞内において同じ方向であるチューブの動径方向に振動する in-phase なモードであるのに対して、高周波 A_{1g} モード (図 3.11(f)) は、A、B の 2 つの原子が反対方向に振動する out-of-phase な振動である為に起こる。

高周波数帯ラマン活性モード 3 つ (図 3.11 の (e)、(f)、(g)) はすべて out-of-phase なモードであるのに対して、低周波数帯ラマン活性モード 4 つ (図 3.11 の (a)、(b)、(c)、(d)) はすべて in-phase なモードであることがわかる。また、図 3.11(e)、(g) のモードは、グラファイトのラマン活性モードである図 3.3(c) の 1588.8cm^{-1} にある E_{2g} モードと、三つの最近接原子の中の一つ伸縮振動するという点で振動が似ている。この図 3.11(e)、(g) の E_{1g} 、 E_{2g} モードが強いラマン強度を与えることは、グラファイトのラマン活性モードである E_{2g} モード (章 3.1.1 の図 3.3) からきている。また、この高周波数帯ラマン活性モード 3 つが out-of-phase なモードであるということは VH 測定においても強い強度が表れることを示している。

次に、高周波数帯ラマン活性モードである A_{1g} モードは、グラファイトのもう一つの高周波数帯のモードである E_{2g} (3.3(b)) モードの振動をおり曲げた振動であることがわかる。しかしながら、高周波数帯ラマン活性モードである A_{1g} モードは、図 3.11(f) と次の図 3.15(g) を比べてわかるように、armchair 型と zigzag 型では、同じ out-of-phase なモードであるが、異なった方向に振動している。zigzag 型だと、C=C ボンドは、ナノチューブの軸方向に伸縮振動しているのに対して、armchair 型だと、C=C ボンドは、ナノチューブの軸方向と垂直な方向に伸縮振動している。これらのことからナノチューブの円曲表面の影響を受けることがわかり、ナノチューブの螺旋度に依存していることが予想される。

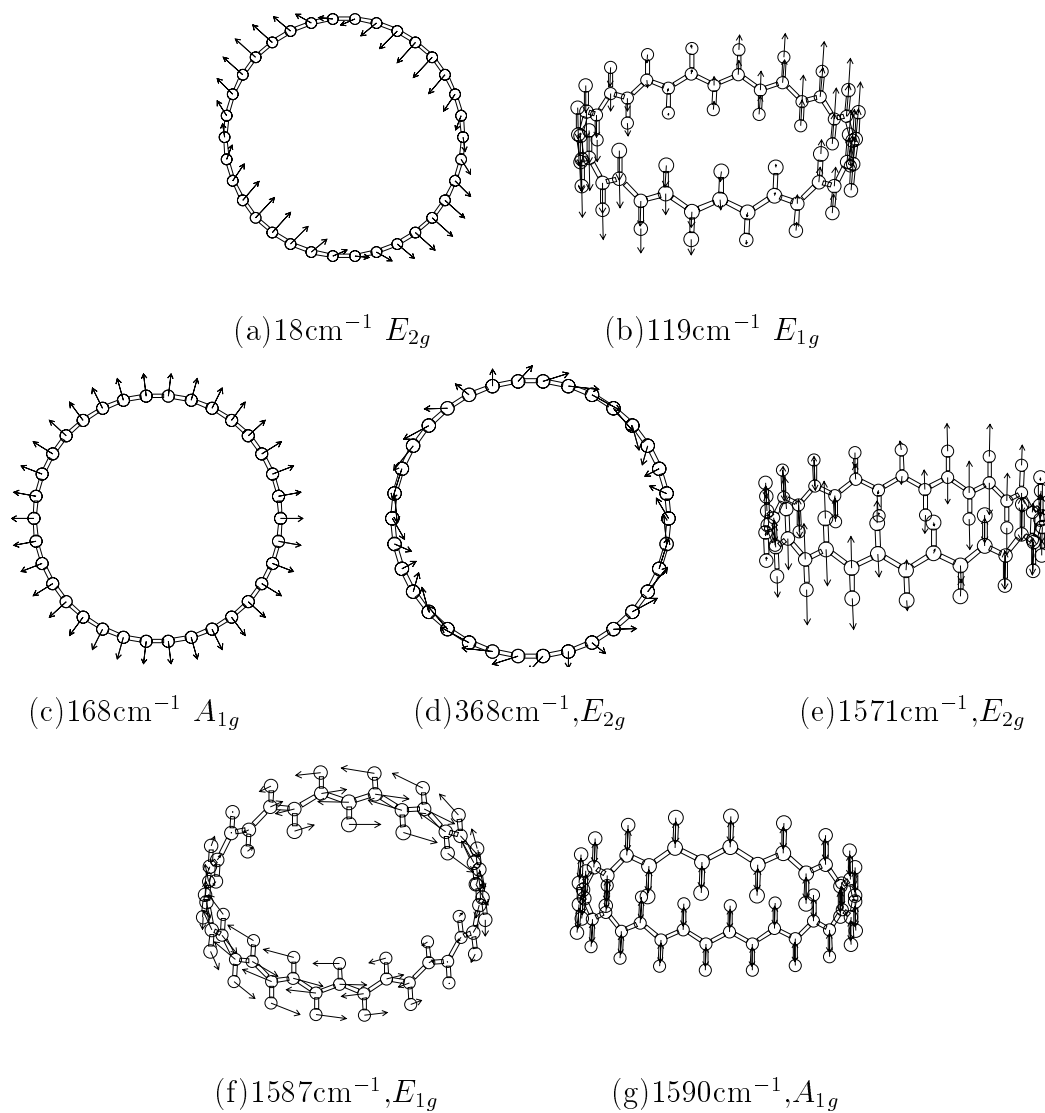


図 3.15. (17,0) ラマン活性モード

図 3.16 に armchair 型と呼ばれる、いわゆる螺旋度のないナノチューブの高周波ラマン活性周波数の半径依存性を、横軸、縦軸にそれぞれ、ナノチューブの半径 r (), 周波数 $\omega(\text{cm}^{-1})$ をプロットして示す。

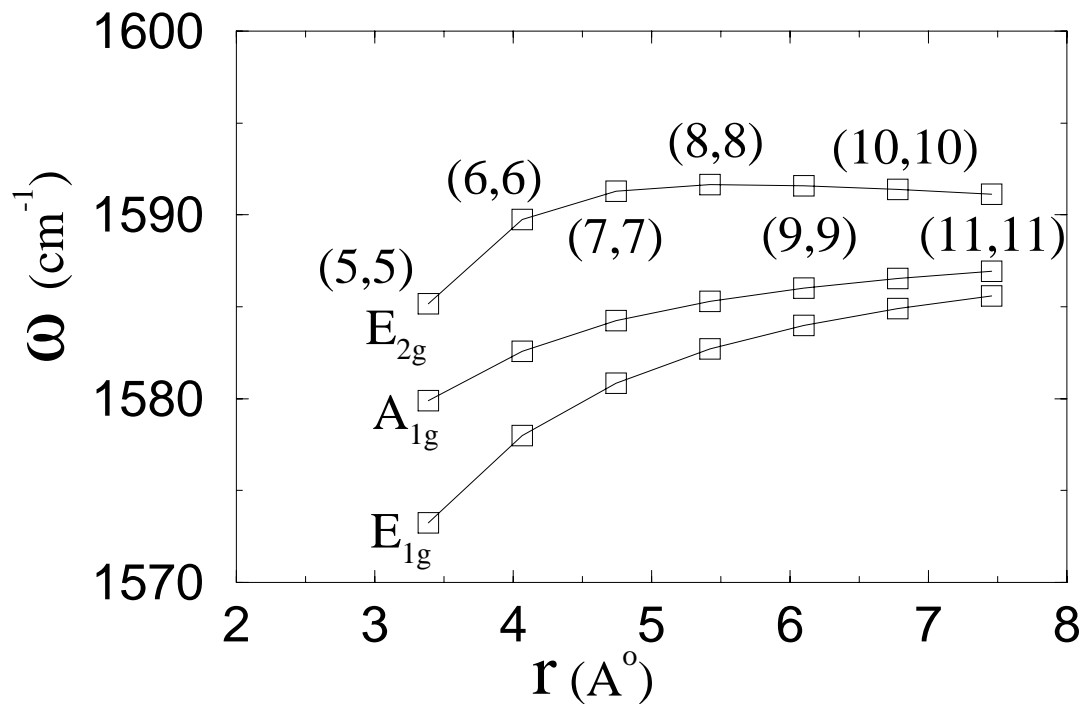


図 3.16.armchair 型ナノチューブの高周波ラマン活性周波数の半径依存性

Kasuya らの実験において、高周波ラマン活性周波数は、ナノチューブの半径が小さくなるとソフト化することが報告されているが 3.16 を見てわかるように、例えば、半径 $r = 6.78$ (), 螺旋度 (10,10) のナノチューブの場合 $\Delta = E_{2g} - E_{1g} \approx 6\text{cm}^{-1}$ 程度であるが、半径 $r = 3.39$ (), 螺旋度 (5,5) のナノチューブの場合 $\Delta = E_{2g} - E_{1g} \approx 13\text{cm}^{-1}$ 程度となり、明らかにソフト化していることがわかり、チューブの円筒面の影響を受けているのがわかる。

また、高周波数帯ラマン活性モード 3 つの強度は図 3.16 の様に螺旋度 (10,10) のナノチューブだと、数 cm^{-1} 程度の非常に近い範囲に表れて実験では分離することが難しいけれども、次の章 3.2.2 で示すナノチューブのラマン強度の角度依存性を調べることによって分離できる。

3.2.2 カーボンナノチューブのラマン強度の角度依存性

図 3.17に螺旋度 (10,10) のナノチューブのラマン強度の角度依存性を示す。

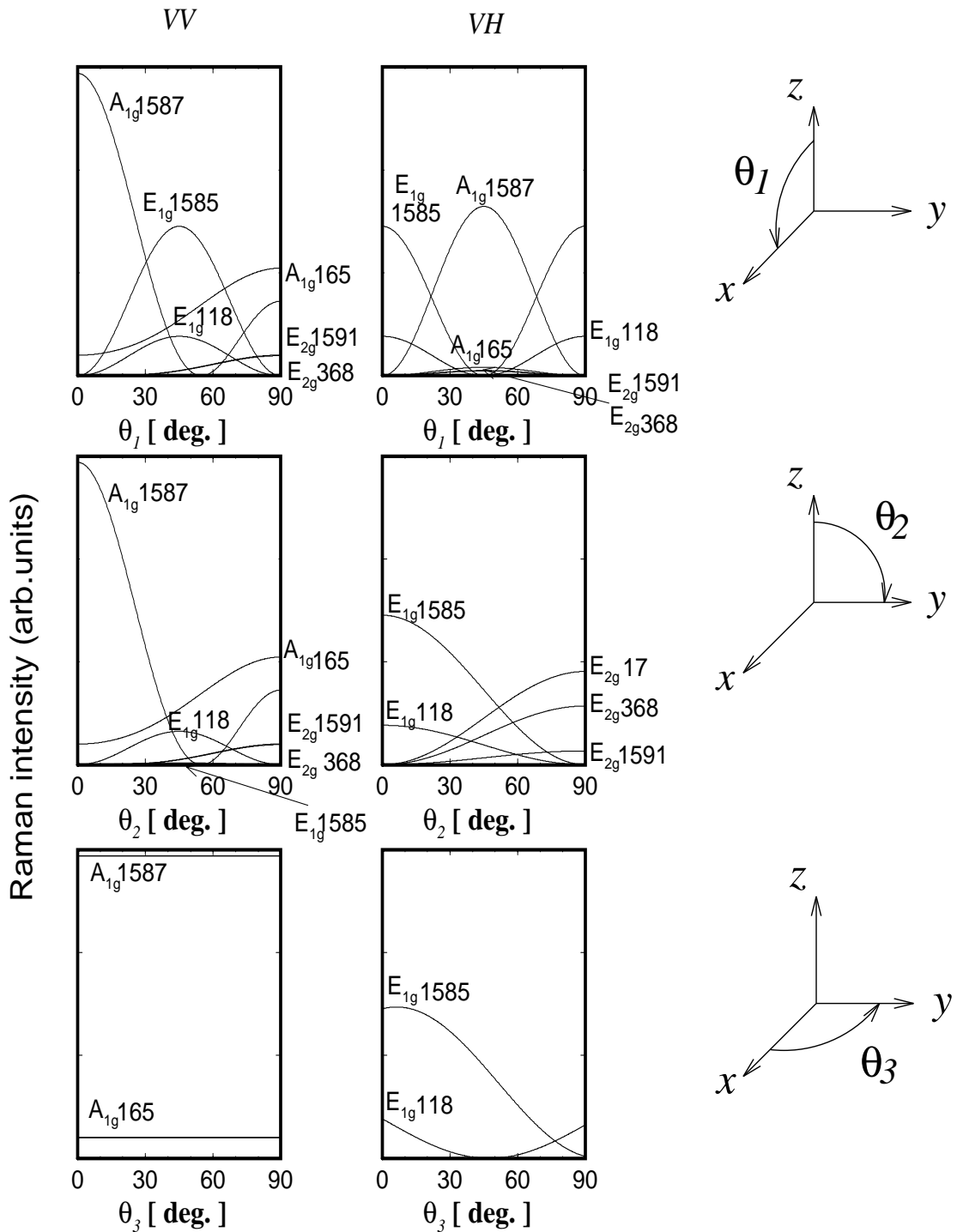


図 3.17. 螺旋度 (10,10) のナノチューブのラマン強度の角度依存性

図 3.17は螺旋度 (10,10) のチューブの軸を z 軸方向に置き、光の分極方向を V(z 方向)、H(y 方向) と固定し、VV 測定とは入射光、散乱光の分極方向がそれぞれ z 、 z 方向で

平行な場合の測定、VH 測定とは入射光、散乱光の分極方向がそれぞれ z 、 y 方向で垂直な場合の測定をさし、 θ_1 、 θ_2 、 θ_3 はチューブの軸が z 軸方向にあるナノチューブをそれぞれ z 軸から x 軸への y 軸回転、 z 軸から y 軸への x 軸回転、 x 軸から y 軸への z 軸回転であり各 $0^\circ \leq \theta \leq 90^\circ$ まで回転させた時のラマン強度の各角度における依存性を横軸、縦軸にそれぞれ、各回転角度 θ 、ラマン強度を示す。ここで、各角度における $\theta=0^\circ$ はナノチューブの軸が z 軸方向にあり、かつ炭素原子が x 軸上にある時である。また各 $theta=0^\circ$ における各モードのラマン強度は一致する。

図 3.17 の VV 測定の θ_1 回転において $\theta_1=0^\circ$ における強度最大値は、 $\omega=1587\text{cm}^{-1}$ の高周波 A_{1g} であり、 $\omega=1585\text{cm}^{-1}$ の高周波 E_{1g} の強度は表れないが、 $\theta_1=45^\circ$ では、逆に $\omega=1585\text{cm}^{-1}$ の高周波 E_{1g} が強度最大値となっている。一方、逆に $theta_1$ 回転の VH 測定では、 $\theta_1=0^\circ$ における強度最大値は $\omega=1585\text{cm}^{-1}$ の高周波 E_{1g} であり、 $\omega=1587\text{cm}^{-1}$ の高周波 A_{1g} の強度は表れないが、 $\theta_1=45^\circ$ では、逆に $\omega=1587\text{cm}^{-1}$ の高周波 A_{1g} が強度最大値となり、 $\omega=1585\text{cm}^{-1}$ の高周波 E_{1g} における強度が表れない。また VH 測定の $theta_2$ 回転における $\theta_2=90^\circ$ に 3 つの高周波ラマン活性モードの内表れるのは $\omega=1591\text{cm}^{-1}$ の E_{2g} だけである。このように章 3.2.1 の図.3.10 で示したランダムな場合では分けられない高周波領域ラマン活性モードの 3 つの周波数をナノチューブのラマン強度の角度依存性を調べることによって分離できる。次に表.3-1 に各角度 $\theta_n(n = 1, 2, 3)$ に表れる高周波領域ラマン活性モードを示す。

VV				VH			
θ_n	0°	45°	90°	θ_n	0°	45°	90°
θ_1	A_{1g}	(E_{1g}, A_{1g})	(A_{1g}, E_{2g})	θ_1	E_{1g}	(A_{1g}, E_{2g})	E_{1g}
θ_2	A_{1g}	(A_{1g}, E_{1g})	(A_{1g}, E_{2g})	θ_2	E_{1g}	(E_{1g}, E_{2g})	E_{2g}
θ_3	A_{1g}	A_{1g}	A_{1g}	θ_3	E_{1g}	E_{1g}	E_{1g}

表.3-1

また、 $\omega=1587\text{cm}^{-1}$ の高周波 E_{2g} モードと $\omega=368\text{cm}^{-1}$ の低周波 E_{2g} モードの 2 つの E_{2g} モードは VV 測定ではほとんど同じ強度を示すが、VH 測定の θ_2 特性においては異なった依存性を示す。

3.2.3 フォノン状態密度とラマン強度

図 3.10 の 3 つのナノチューブの単位胞内の原子数は螺旋度 (10,10), (17,0), (11,8) それぞれ、40、68、364(個)である。また強いピークが出るモードは章 3.2.1 で述べたように螺旋度や半径によらず 7(個)であるがラマン活性モードの全フォノン数に占める割合は、螺旋度 (10,10)、(17,0)、(11,8) それぞれ $7/120$ 、 $7/204$ 、 $7/1092$ で大幅に違う。図 3.18 に横軸、縦軸にそれぞれ、ナノチューブの原子数 $N(\text{C-atom})$ 、ブリージングモード (A_{1g}) 強度をとり、ナノチューブの原子数に対してラマン強度がどのように変化するか示した図である。また、印は、螺旋度の違うナノチューブ (n, m) ($n=10, 1 \leq m \leq 10$) のブリージングモードのラマン強度の強さを示し、印は、螺旋度 (10,10) のナノチューブの単位胞内をそれぞれ 1、2、4、6、8、10 個並進方向につなげた時の強度を示す。

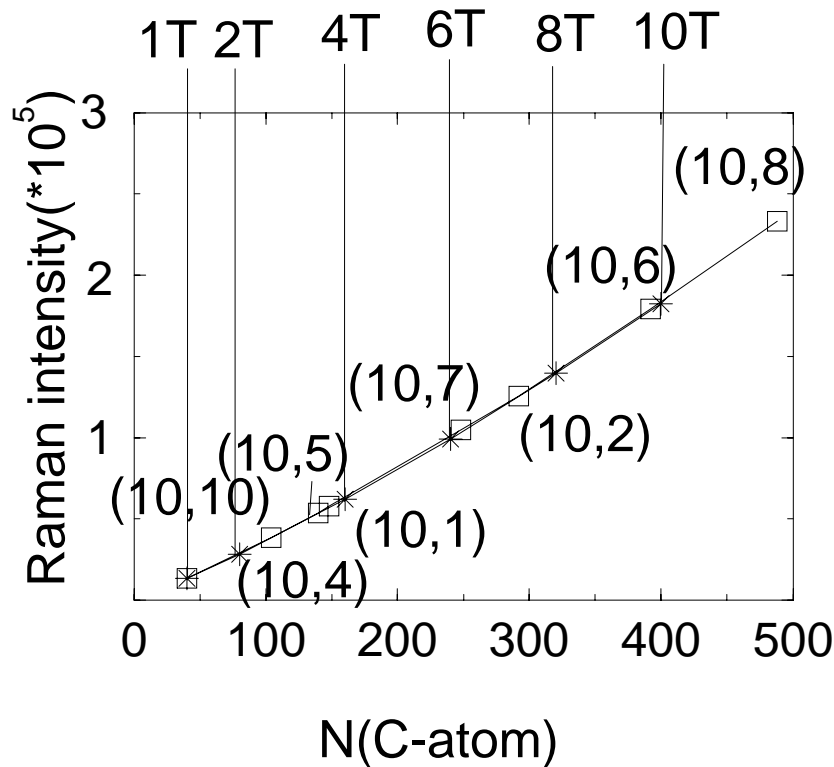


図.3.18 ブリージングモード (A_{1g}) のラマン強度のピークの強さの炭素原子数依存性

フォノン状態密度より、ラマン強度の強さは、ナノチューブの単位胞内の原子数 N に反比例すると思われるが、今回計算した結果では、ナノチューブの単位胞内の原子数 N にラマン強度の強さ (I) は反比例してなく原子数 N に比例していることがわかった。図 3.18 より、炭素原子 1 個あたりのブリージングモードラマン強度の強さが求ま

る。

$$I/N = 3.5775 \times 10^{-2} / (C - atom) \quad (3.2)$$

となる。例えば、螺旋度 (10,10)、原子数 $N=40$ のナノチューブと、螺旋度 (10,8)、原子数 $N=488$ のナノチューブとのブリージグモード (A_{1g}) のラマン強度のピークの強さを比べた場合

(螺旋度 (10,10) の強度のピークの強さを 1.0 とした場合)

(例)

螺旋度	I	螺旋度	I
(10,10)	1.0	(10,8)	16.03

となり、螺旋度 (10,10) のチューブに比べ、螺旋度 (10,8) のナノチューブのブリージグモードにおける強度は約 16 倍の強度が得られる。

3.2.4 カーボンナノチューブのラマン強度の端の効果

章 3.2.1では、端の無い固体のナノチューブのラマン強度を計算した (図 3.10)。しかし実験では、章 1.6の螺旋度 (10,10) の単相カーボンナノチューブ (SWCN) のラマン強度の実験値 [2] の様に中間周波数領域では微弱ながらもラマン強度があるのにも関わらず、理論的な計算結果では固体のチューブでは、全く強度が存在しなかった。今回、中周波数領域におけるラマン強度はナノチューブの端に寄与していると考え、この章では、端を持つ単層カーボンナノチューブのラマン強度を計算する。

図 3.19は、螺旋度 (10,10) のナノチューブを並進方向 T (ナノチューブの並進ベクトル 章 1.3.3. 参照) に $20T$ (長さ $L_{20T}^{(10,10)} = 49.19(\text{Å})$) つなげたナノチューブの軸方向ガンダムにある場合の端効果を考慮したラマン強度を立体角で積分したものである (縦軸、横軸は、それぞれ、ラマン強度、ラマンシフトを示す)。ラマン強度の測定方法であるが、これは章 3.2.1の測定方法と同じで、 VV 測定、 VH 測定測定を考える。 VV 測定とは、入射光、散乱光の分極方向がそれぞれ平行、 VH 測定とは、入射光、散乱光の分極方向がそれぞれ垂直な場合の測定方法である。

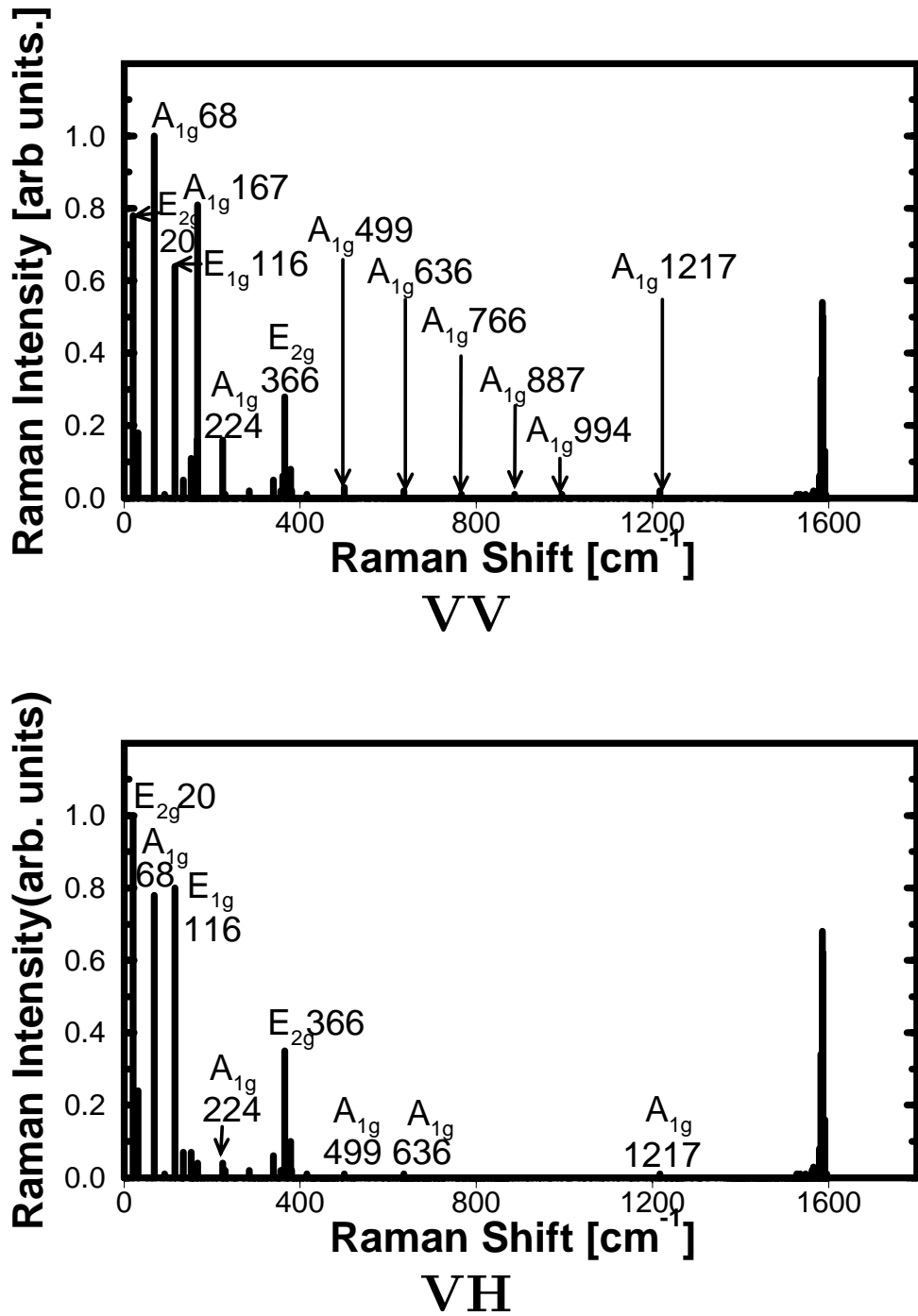


図 3.19. 端のある螺旋度 (10,10)、長さ $L_{20T}^{(10,10)} = 49.19()$ のチューブのラマン強度

章 3.2.1の図 3.10 と図 3.19を比べてわかるように、端の無い固体のナノチューブのラマン強度に見られない中間周波数領域におけるラマン強度が、図 3.19では、微弱ながらもラマン強度が表れる。ここで、注目してもらいたいのは、端効果を考慮したラマン強度の活性モードには、端がない固体のナノチューブには見られないモードが表れる点である。中間周波数領域におけるラマン活性モードは、端の無い固体のナノチューブ

ブのラマン活性モードには存在しないナノチューブの軸方向のモードや、端に局在したモードに対応している。有限の長さの効果は、低周波数領域にもブリージングモードの変化を与える。以下、(A) 低周波数領域のブリージングモードの変化、(B) にナノチューブの軸方向のモード (C) に端に局在したモードを示し、それぞれのモードについて考察する。

(A) ブリージングモード (A_{1g}) の変化

図 3.21 に、端の効果による動径方向のブリージングモードのモードの変化を示す。ここで、図 3.21 の各 (a) ~ (f) において左図は、各周波数における振動モード、右図は、各長さにおける原子の振動方向の大きさを、縦軸に長さ ()、横軸に、図 3.20 の様に

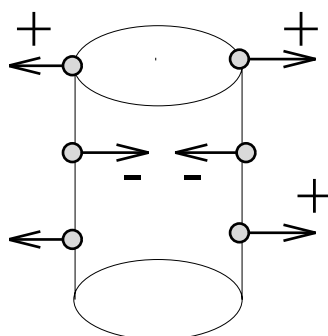


図 3.20. 有限の長さによるブリージングモードの振動変化

チューブが動径方向に広がる時 (チューブの中心方向と反対向き) を +、縮まる時 (チューブの中心方向) を - とし、各長さにある原子の固有ベクトルの大きさの和を示す。

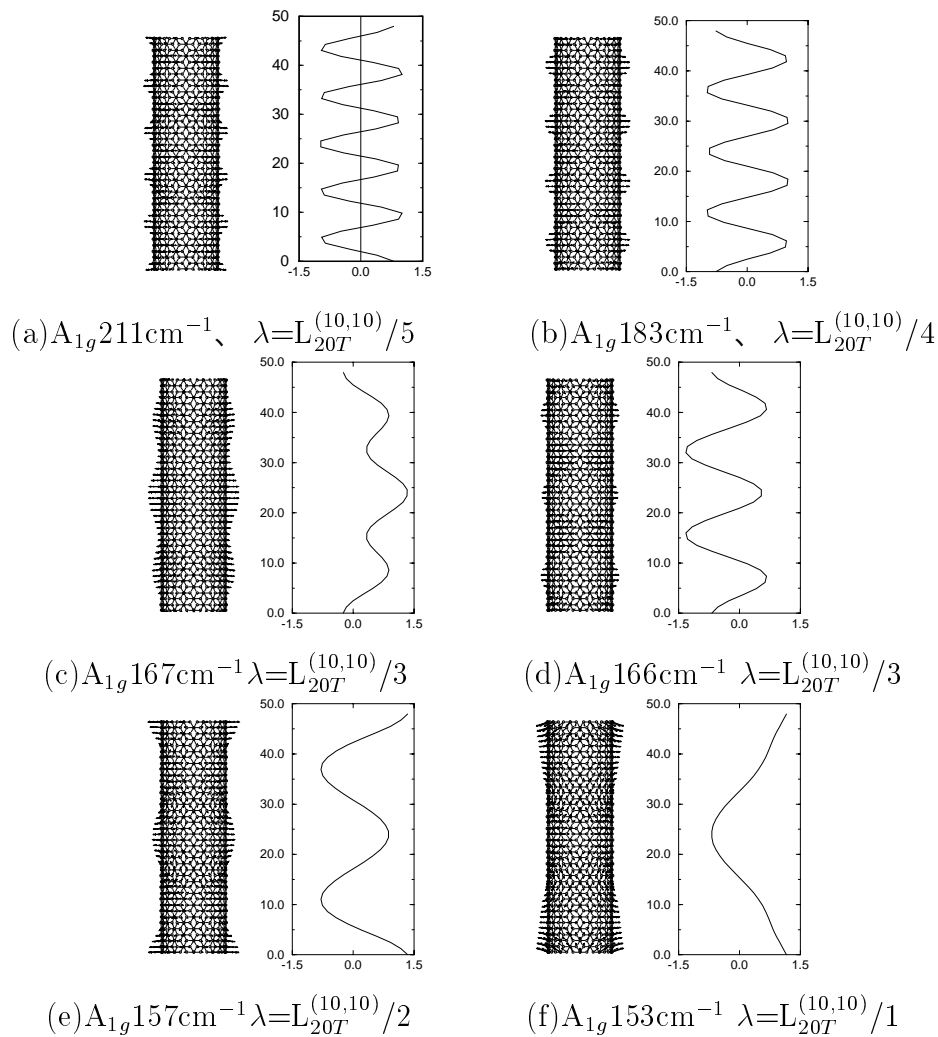


図 3.21. A_{1g} (ブリージング) モードの変化

ここで、ブリージングモードは、それぞれ、(a) $\lambda=L_{20T}^{(10,10)}/5$ 、(b) $\lambda=L_{20T}^{(10,10)}/4$ 、(d) $\lambda=L_{20T}^{(10,10)}/3$ 、(e) $\lambda=L_{20T}^{(10,10)}/2$ 、(f) $\lambda=L_{20T}^{(10,10)}/1$ の周期で広がったり、縮んだりする。図 3.21 の 6 つのブリージングモードの中でラマン活性モードであるが、軸方向に沿って、この振幅の大きさを積分した時、+、- 成分のどちらかが、大きい時、ラマン活性モードとなり、ラマン強度が表れる。また、+、- 成分のどちらかが、大きければ大きいほど強いラマン強度を示す。この、中でのラマン活性モードであるが、上の理由により、(c)、(d)、(e)、(f) がラマン活性モードとなり、VV 測定ではラマン強度を表す。これらのブリージングモードも、端を考えない固体のナノチューブのブリージングモード(章の図を参照)と同様に、VV 測定では強いラマン強度をしめすが、VH 測定では、ラマン強度が非常に弱くなる。この 6 つのブリージングモードのなかで、最も強いラマン強度を示すのが、(c) の $A_{1g} 167\text{cm}^{-1}$ 、 $\lambda=L_{20T}^{(10,10)}/3$ のモードであるが、図 3.21 の (c)

を見てわかる様に、+ 成分の振動が非常に大きくなっていて、- 成分の振動は、ナノチューブの端付近のの原子のみである。また、 λ の周期が周期が短くなれば、+、- 成分がほぼ等しくなり、ラマン活性モードとはならない。

(B) ナノチューブの軸方向のモード

VV、VH 測定の中周波数領域 ($500\text{cm}^{-1} \leq \omega < 1500\text{cm}^{-1}$) におけるラマン強度のラマン活性モードは、図 3.23 の (b) ~ (f) の様にナノチューブの軸方向のモードである。今、図 3.23 の各 (a) ~ (f) の左図は、各周波数における振動モード、右図は、各長さにある原子の振動方向の大きさを、縦軸ナノチューブにおける原子の位置を ()、横軸に、図 3.22 の様に

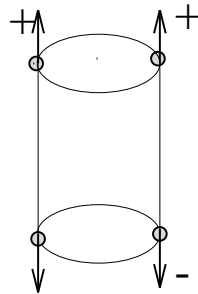
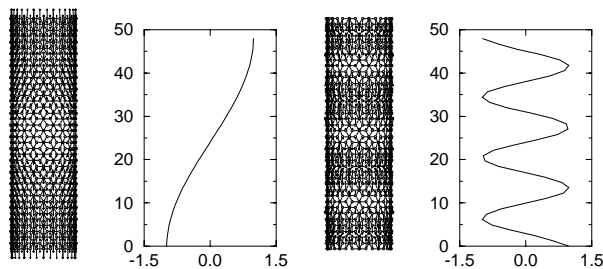


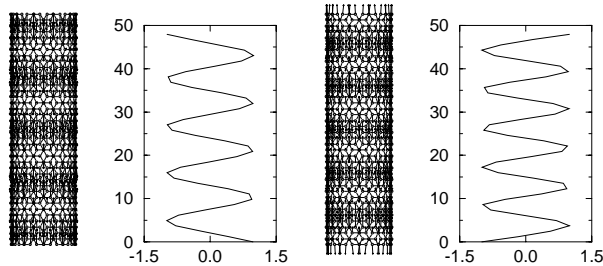
図 3.22. 有限の長さによるチューブの軸方向の振動方向

ナノチューブの軸方向に +、- 方向をさだめ、各長さにある炭素原子の固有ベクトルの大きさの和を示す。



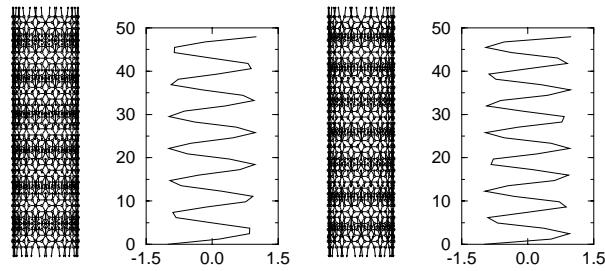
(a) A_{1g} 、 68cm^{-1} 、 $\lambda=L_{20T}^{(10,10)}/(1/2)$

(b) A_{1g} 、 499cm^{-1} 、 $\lambda=L_{20T}^{(10,10)}/3$



(c) A_{1g} 、 636cm^{-1} 、 $\lambda=L_{20T}^{(10,10)}/4$

(d) A_{1g} 、 766cm^{-1} 、 $\lambda=L_{20T}^{(10,10)}/5$



(e) A_{1g} 、 887cm^{-1} 、 $\lambda=L_{20T}^{(10,10)}/6$ (f) A_{1g} 、 994cm^{-1} 、 $\lambda=L_{20T}^{(10,10)}/7$

図 3.23. ナノチューブの軸方向のモード

中周波数領域 ($500\text{cm}^{-1} \sim 1500\text{cm}^{-1}$) におけるラマン強度は、ナノチューブの軸方向に寄与したモードである。図 3.23 の (d)、(e)、(f) の 3 つの A_{1g} モードは、VV 測定での中周波数領域におけるラマン強度において微弱ながらも強度を表す。また VH 測定では、動径方向ブリージングモードと同様に、非常に小さくなり、中周波数領域において、ラマン強度を示さない。(d)、(e)、(f) それぞれ、 $\lambda=L_{20T}^{(10,10)}/5$ 、 $\lambda=L_{20T}^{(10,10)}/6$ 、 $\lambda=L_{20T}^{(10,10)}/7$ 、の周期で、ナノチューブの軸方向に振動している。

しかしながら、図 3.23(b)、(c) のモードは VH 測定でも、VH 測定より強度は小さくなるが、強度を示す。これは、付録 A1 に図 3.23(c) と図 3.23(d) の各長さにある炭素原子の固有ベクトルの和を示すが、図 3.23(c) 636cm^{-1} の方が 3.23(d) 766cm^{-1} よりも端付近の原子のナノチューブの軸方向の振動の周期がずれ大きくなっていることに関係していると思われる。例えば付録 A の様にナノチューブの midpoint から一方の端迄の z 軸方向の固有ベクトルの成分の和 S をとった結果 (midpoint に対して対称の振動をしているので一方のみでよい。)、図 3.23 の (a)、(b)、(c)、(d) 各 $S=12.6080$ 、 -1.82200 、 1.42200 、 1.18400 より、短波長になればなるほどナノチューブの midpoint に対して伸縮する振動の大きさが減り、これが VH 測定で得られない原因ではないかと思われる。

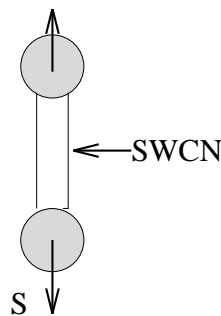


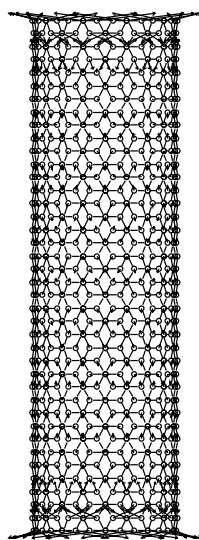
図 3.24.

例えば、今図 3.24 の様に SWNC (単相カーボンナノチューブ) をボンドと仮定すると、あたかも端のないナノチューブの高周波数ラマン活性モードである A_{1g} モード (章 3.2.1 の

図 3.11) のボンド間における振動と似ている。また、図 3.23 の (a) の A_{1g} 、 68cm^{-1} の z 軸方向のブリージングモードは VV 測定においては、図 3.21(c) のモードよりも強い強度を示し、最も強いラマン強度を表す。チューブを軸方向に引っ張る力が強いために 3.21(c) のブリージングモードは、VH 測定では非常に小さくなるのに対して、図 3.23 の (a) のモードは強度は小さくなるが強い強度を示す。

(C) その他の端の効果によるラマン活性モード

その他の端に局在したラマン活性モードの一例として、図 3.25 の A_{1g} 、 1217cm^{-1} のモードをあげる。

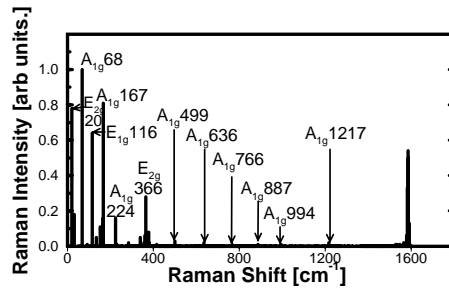


A_{1g} 、 1217cm^{-1}

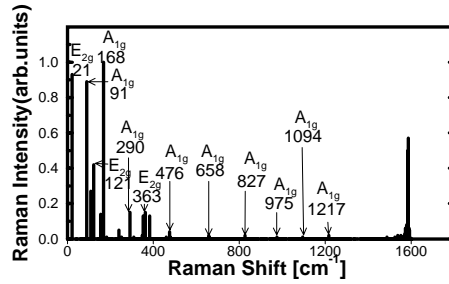
図 3.25. 端に局在したラマン活性モード

中周波数領域では、(B) で述べたナノチューブの軸方向のモードの他に、図 3.25 のような、端付近の原子のみが大きく振動していて、端付近においてちょうどブリージングモード (章 3.2.1 の図 3.13 参照) のような振動をしている。このモードは明らかに端の効果を受けた振動であることがわかる。このモードは VV、VH 測定でも微弱ながら、計算で得られる。

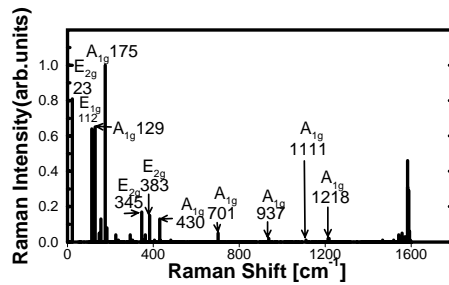
端を持つナノチューブの長さに対するラマン強度の変化を VV 測定の場合について図 3.26 に示す。



(a) 20T ($L_{20T} = 47.96 [\text{\AA}]$)



(b) 15T ($L_{15T} = 35.66 [\text{\AA}]$)



(c) 10T ($L_{10T} = 23.36 [\text{\AA}]$)

図 3.26. 端を持つ螺旋度 (10,10) のナノチューブの長さに対するラマン強度の変化 (VV)

ここで、(a), (b), (c) は T を章 1.3.3 で示した、螺旋度 (10,10) ナノチューブのユニットセルの並進方向の長さとする、各 20, 15, 10 個のユニットセルをつなげた端を持つナノチューブの VV 測定ラマン強度であり、長さは各 $L_{20T} = 47.96 [\text{\AA}]$, $L_{15T} = 35.66 [\text{\AA}]$, $L_{10T} = 23.36 [\text{\AA}]$ である。また (a), (b), (c) 各強度の相対比は 1:0.77:0.56 となっている。(a), (b), (c) の各長さのチューブとも中間周波数領域に離散的なラマン強度が現れている。この中間周波数領域のナノチューブの長さに対するラマン活性周波数の依存性を次の図 3.27 に縦軸、横軸にそれぞれ、周波数 $\omega [\text{cm}^{-1}]$, 各 20T, 15T, 10T のナノチューブの長さ $L [\text{\AA}]$ をプロットして示す。

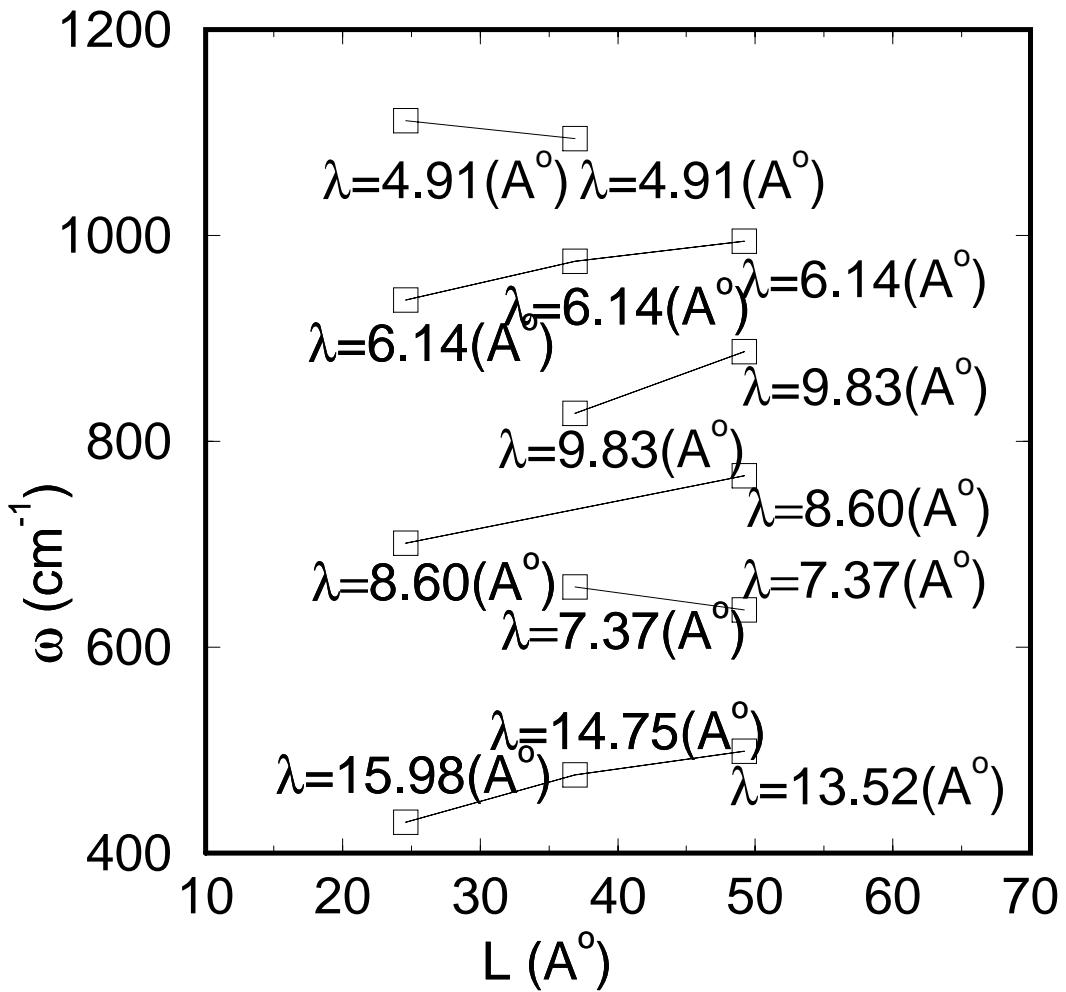


図 3.27. 中間周波数領域のラマン活性周波数のチューブの長さ依存性

図 3.27を見るとわずかながらであるがラマン活性周波数の長さ依存性が見られる。また、中間周波数領域に見られるラマン強度は、 z 軸方向の振動であるが、ラマン強度として現れるモードは、 400cm^{-1} 付近のモードを除いては、同じ周期的振動をするモードである。このことは、中間周波数領域のラマン強度が離散的に現れることを示している。この結果は、A. M. Rao[2] から図 1.5の実験値とも一致している。

図 3.25の端の原子のみが振動するモードであるが、ナノチューブの長さには寄与しない端の原子のみが半径方向に振動するモードであるため長さ依存性は見られない。例えば、20T, 15T, 10T のナノチューブだと、それぞれ、 $1217, 1217, 1218\text{cm}^{-1}$ でほとんど変化は見られない。

図 3.23(a) の低周波数領域に現れるナノチューブの軸方向のブリージングモードのチューブの長さの依存性であるが、明らかに依存性があり、ナノチューブの長さが長くなるにつれて、低周波数領域側にシフトしていつているのがわかる。またこのモードは、

端を持つラマン強度の計算結果では非常に強い強度を示す。

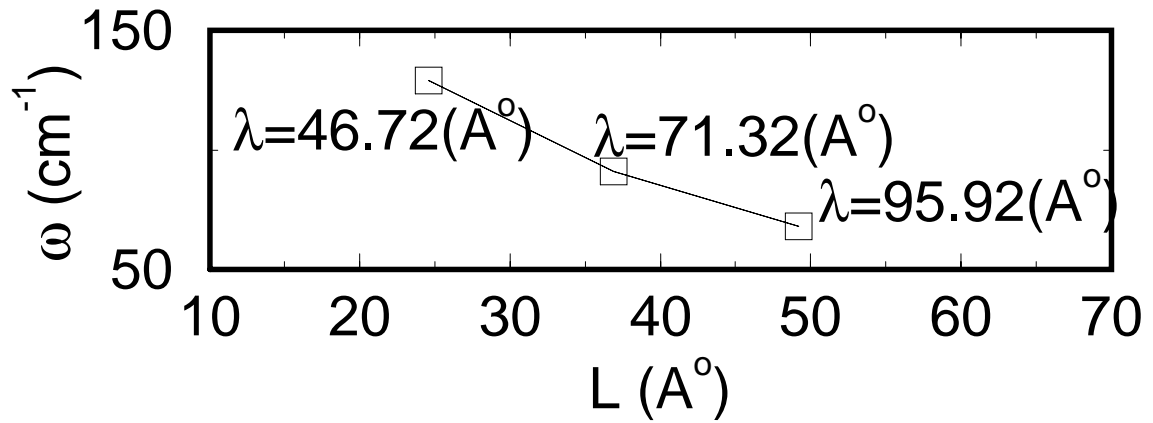


図 3.28. z 軸方向のブリージングモードの周波数の長さ依存性

また、長さがなればある一定の周波数で収束するのではないかと思われる。

第 4 章

結論

本研究では単層カーボンナノチューブ (SWCN) のフォノン分散関係とラマン強度を計算するプログラムを開発した。得られた結論として (1) SWCN の音速は、縦波、ねじれ波、横波の順で速い。ここでねじれ波とは円筒系の SWCN をねじりながら進む波である。横波が 2 重に縮重するので、SWCN には、4 つの音響モードが存在し、各 $10^{-2}km/s$ のオーダーで螺旋度に依存している。(2) SWCN の低振動数ラマンモード ($500cm^{-1}$ 以下) の振動数は、半径のみに依存し螺旋度には依存しない。したがってその依存性より振動数から直接半径を推定できる。(3) SWCN の高振動数ラマンモード ($1590cm^{-1}$ 付近) の振動数は基本的にグラファイトのラマンモードを折り返したモードである。この振動数もわずかながら半径の依存性がある。(4) 端のない SWCN の中間振動数ラマンモード ($500-1250cm^{-1}$) の振動数はラマン強度が無かった。しかし SWCN を有限の長さにすると強度が現れる。(5) ラマン強度の角度依存性を求めてみることによって、実験から直接ラマンモードの対称性を求めることができる。この結果はとくに高振動数ラマンモードの分離に有効である。

参考文献

- [1] R. A. Jishi, L. Venkataraman, M. S. Dresselhaus and G. Dresselhaus, Phys. Chem. Lett, **209**, 7, (1993.)
- [2] A. M. Rao, E. Richer, Shunji Bandow, Bruce Chase, P. C. Eklund, K. A. Williams, S. Fang, K. R. Subbaswamy, M. Menon, A. Thess, R. E. Smalley, G. Dresselhaus and M. SDresselhaus, Science, **275**, 187, (1997.)
- [3] S. Guha, J. Menendez, J. B. Page and G. B. Adams, Phys. Rev. B, **53**, 13016, (1996.)
- [4] R. saito, G. Dresselhaus and M. S. Dresselhaus. Phys. Rev.B, **53**, 2044, (1995.)
- [5] カーボンナノチューブ 飯島澄男 固体物理 **27**, 442, (1992.)
- [6] フラーレン及びカーボンナノチューブの数学と電子構造. 齊藤理一郎
- [7] C_{60} フラーレンの化学 「化学」編集部編 化学同人
- [8] Shunji Bandow, A. M. Rao, K. A. Williams, A. Thess, R. E. Smalley, and P. C. Eklund J. Phys. Chem. B , **101**, 8839, (1997.)
- [9] Carbon NanoTubes Preparation and Properties. Thomas W. Ebbesen.
- [10] 第12回フラーレン総合シンポジウム講演要旨集、フラーレン研究会
- [11] H. Kataura et al, unpublished
- [12] T. Guo, C. M. Jin, and R. E. Smalley, Chem. Phys. Lett, **234**, 49-54, (1995.)

-
- [13] A. Thess, R. Lee, P. Nikolaev, H. Dai, Petit, J. Robert,
C. Xu, Y. H. Lee, S. G. Kim, A. G. Rinzler, D. T. Colbert, G. E. Scuseria,
D. Tomanek, J. E. Fischer, and R. E. Smalley,
Science **273**, 483, (1996.)
- [14] D. W. Snoke and M. Cardona, *Solid State Commun*, **87**, 121, (1993).

付録 A

データ

A.1 ナノチューブの端ノ効果による軸方向のモードのデータ

以下にナノチューブのラマン強度の考察のために、軸方向の振動データモードのデータを示す。

636cm, A1g	
L()	eginvector(z 方向)
0.00000	0.97800
1.22976	0.42400
2.45951	-0.19000
3.68927	-0.78200
4.91902	-0.98400
6.14878	-0.71600
7.37854	-0.11600
8.60829	0.55400
9.83805	0.94800
11.06780	0.89000
12.29756	0.41400
13.52732	-0.27400
14.75707	-0.81800
15.98683	-0.97600
17.21659	-0.67000
18.44634	-0.03200
19.67610	0.60800
20.90585	0.96600
22.13561	0.86200
23.36537	0.33600
ナノチューブ中点	
24.59512	-0.33600
25.82488	-0.86200
27.05463	-0.96600
28.28439	-0.60800
29.51415	0.03200
30.74390	0.67000
31.97366	0.97600
33.20341	0.81800
34.43317	0.27400
35.66293	-0.41400

36.89268	-0.89000
38.12244	-0.94800
39.35219	-0.55400
40.58195	0.11600
41.81171	0.71600
43.04146	0.98600
44.27122	0.78200
45.50097	0.19000
46.73073	-0.42400
47.96049	-0.97800

766cm, A1g

L()	eginvector(z 方向)
0.00000	-0.99200
1.22976	-0.16400
2.45951	0.56600
3.68927	0.99200
4.91902	0.70400
6.14878	-0.06600
7.37854	-0.79400
8.60829	-0.96600
9.83805	-0.46400
11.06780	0.36200
12.29756	0.93400
13.52732	0.85600
14.75707	0.17800
15.98683	-0.62200
17.21659	-0.99000
18.44634	-0.66600
19.67610	0.12400
20.90585	0.82600
22.13561	0.95200
23.36537	0.41400
ナノチューブ中点	
24.59512	-0.41400
25.82488	-0.95200
27.05463	-0.82600
28.28439	-0.12400
29.51415	0.66600
30.74390	0.99000
31.97366	0.62200
33.20341	-0.17800
34.43317	-0.85600
35.66293	-0.93600
36.89268	-0.36200
38.12244	0.46400
39.35219	0.96600
40.58195	0.79400
41.81171	0.06600
43.04146	-0.70600
44.27122	-0.99200
45.50097	-0.56600
46.73073	0.16400
47.96049	0.99200

A.2 データの場所

1. チューブの固有値、固有ベクトル = /takeya/for/tube/raman/ にある。
(a)n-m.dat2= 螺旋度 (n,m) のチューブの固有値、固有ベクトル (b)n-m.xyz= 螺旋度 (n,m) のチューブの xyz(xmol) (c)n-m.nera= 螺旋度 (n,m) のチューブの最近接

付録 B

プログラムソース

以下に、ナノチューブのフォノン分散関係とラマン強度を計算するために著者が開発したプログラムを示す。

B.1 円筒形の座標を求めるプログラムソース

場所:

file= takeya/for/tube/tube-xyz.f

1. 入力:

プログラムを実行すると、display 上に 'カイラルベクトル $C_h =$ ' が現れるので、2つのカイラルベクトルを display 上に $n\ m$ を入力すれば良い。

2. 出力ファイル:

(1) tube.xyz = チューブのユニットセルの xmol 用座標

(2) en.xyz = phonon と 最近接原子を求めるために必要なチューブの xyz 座標

(3) t-ch = チューブの phonon を求めるために必要なパラメータの file

```
c
c
c 円筒形の座標を求めるプログラム
c
c xmol 及び tube phonon 出力用
c
c file= ~takeya/for/tube/tube-xyz.f
c
c ファイルからの入力 (FILE NAME= C_h) C_h (n,m)
c           出力 tube.xyz (xmol 用)
c           en.xyz2 (最近接 用)
c
c programed by takao takeya
c
c date 95.10.31,
c       97.03.03, modified by R. Saito
c
c
c implicit real*8(a-h,o-z)
c parameter (nk=3000)
c dimension x(nk),y(nk),z(nk)
c
```



```
c  acc: C-C 間結合距離
c
c      acc=1.42d0
c
c  ファイルからの入力 (FILE NAME= C_h)
c
c
c      write(*,*) ' カイラルベクトル C_h ='
c      read(*,*) n,m
c
c
c      call gen11(n,m,np,nq,ndr)
c      write(*,*) ' 対称 ベクトル R = (', np, ', ', nq, ', ' )'
c
c  ユニットベクトルの大きさ, 円周率を設定
c  a;|a| , pi; 円周率 sq3 = 3
c
c      sq3=sqrt(3.0d0)
c      pi=4.0d0*atan(1.0d0)
c      a=sqrt(3.0d0)*acc
c
c  対象ベクトル、カイラルベクトル、並進ベクトルの大きさを求める
c  r;|R| , c;|C_h| , t;|T|
c
c      r=a*sqrt(dfloat(np*np+nq*nq+np*nq))
c      c=a*sqrt(dfloat(n*n+m*m+n*m))
c      t=sqrt(3.0d0)*c/dfloat(ndr)
c      write(*,*) 't=',t, dfloat(n*n+m*m+n*m)
c      write(*,190)c/2.0d0/pi
190  format(' 半径 =',f10.4)
c
c  nn: ユニットセル内の六員環の数 N, 炭素原子数 = 2N = 2 * nn
c  rs: 円の半径、を求める
c
c      nn=2*(n**2+m**2+m*n)/ndr
c
c      write(*,*) ' 原子数 =',nn*2
c      if(2*nn.gt.nk) stop 'parameter=nk change'
c
c      rs=c/(2.0d0*pi)
c
c  q1: C_h の カイラル角
c  q2: R の カイラル角
c  q3: C_h と R の なす角
c
c      q1=atan((sq3*dfloat(m))/dfloat(2*n+m))
c      q2=atan((sq3*dfloat(nq))/dfloat(2*np+nq))
c      q3=q1-q2
c
c  q4: A 原子の C_h 上での周期角
c  q5: A 原子と B 原子の C_h 上でのずれの角度
c
c      q4d=abs(r)*cos(q3)/c*2.0d0*pi
c      q4=2.0d0*pi/dfloat(nn)
c      if(abs(q4d-q4).gt.0.0000001) stop 'q4 hen'
c      q5=acc*cos((pi/6.0d0)+q1)/c*2.0d0*pi
c
c  h1: R ベクトルが C_h と 交わるまでの長さ
```

```
c      h2: A 原子と B 原子の z 軸 方向ずれの大きさ
c
      h1=abs(t)/abs(sin(q3))
      h2=acc*sin((pi/6.0d0)+q1)
c      h2=abs(h2)
c
      write(*,*) 'q1: C_h の カイラル角 =',q1*180.0d0/pi
      write(*,*) 'q2: R の カイラル角   =',q2*180.0d0/pi
      write(*,*) 'q4: A 原子の C_h 上での周期角 =',
&q4*180.0d0/pi
      write(*,*) 'q5: B 原子の   周期角 =',q5*180.0d0/pi
c
c
c      ユニットセル内での原子の座標を求める。
c      R ベクトルがとりうる原子 (A 原子) の座標を求める。
c
      ii=0
      do 100 i=0,nn-1
        x1=0
        y1=0
        z1=0
        k=int(dfloat(i)*abs(r)/h1)
        x1=rs*cos(dfloat(i)*q4)
        y1=rs*sin(dfloat(i)*q4)
        z1=dfloat((dfloat(i)*abs(r)-dfloat(k)*h1))*sin(q3)
        kk2=abs(int(z1/t))+1
        if(z1.gt.t-0.02)then
          z1=z1-t*dfloat(kk2)
        endif
        if(z1.lt.-0.02) then
          z1=z1+t*dfloat(kk2)
        endif
        ii=ii+1
        x(ii)=x1
        y(ii)=y1
        z(ii)=z1
c
c      原子上の原子の Z 座標を求める。
c
        z3=(dfloat(i)*abs(r)-dfloat(k)*h1)*sin(q3)-h2
        ii=ii+1
c
c      z3 がユニットセル内にあるかどうか！
c
        if((z3.ge.-0.02).and.(z3.le.t-0.02))then
c
c      ある！
c
          x2 =rs*cos(dfloat(i)*q4-q5)
          y2 =rs*sin(dfloat(i)*q4-q5)
          z2 =dfloat(dfloat(i)*abs(r)-dfloat(k)*h1)*sin(q3)-h2
          x(ii)=x2
          y(ii)=y2
          z(ii)=z2
          else
c
```

```
c   ない!
c
      x2 =rs*cos(dfloat(i)*q4-q5)
      y2 =rs*sin(dfloat(i)*q4-q5)
      z2 =dfloat(dfloat(i)*abs(r)-dfloat(k+1)*h1)*sin(q3)-h2
      kk=abs(int(z2/t))+1
      if(z2.gt.t-0.01)then
        z2=z2-t*dfloat(kk)
      endif
      if(z2.lt.-0.01) then
        z2=z2+t*dfloat(kk)
      endif
      x(ii)=x2
      y(ii)=y2
      z(ii)=z2
      endif
100   continue
c
c   データの出力 xmol 用 (file name=tube.xyz)
c
      open(60,file='tube.xyz')
      write(60,*)2*nn
      write(60,*)' '
      do i=1,nn*2
        write(60,117)x(i),y(i),z(i)
      end do
117   format('C',' ',f25.10,' ',f25.10,' ',f25.10)
      close(60)
c
c   データの出力 phonon & 近接 用 (file name=en.xyz2)
c
      open(60,file='en.xyz2')
      write(60,*)2*nn
      write(60,118)t,acc
118   format(2f25.10)
      do i=1,nn*2
        write(60,116)i,x(i),y(i),z(i)
      end do
116   format(i5,' ',f25.10,' ',f25.10,' ',f25.10)
      close(60)
c
c   データの出力 phonon 用 (file name=t-ch)
c
      open(60,file='t-ch')
      write(60,115)t,c,q4,q5,q1,q3
115   format(5f25.20)
      close(60)
      stop
      end

c
c   新しい R の定義で、tube の parameter を計算する。
c
c   10/16/95   R. Saito
c
      subroutine gen11(n,m,np,nq,ndr)
      dimension nnp(100),nnq(100)
```

```
c
    itest=1
    itest1=0
c
    nd=igcm(n,m)
    if(mod((n-m),3*nd).eq.0) then
        ndr=3*nd
    else
        ndr=nd
    endif
    if(itest.eq.1) nd=nd
    if(itest.eq.1) ndr=ndr
c
    a=sqrt(3.0)*1.421
    eps=1.0e-5
c L
    l2=n*n+m*m+n*m
    if(l2.le.0) stop 'l2.le.0'
    l=int(sqrt(dfloat(l2))+eps)
    dt=a*sqrt(dfloat(l2))/3.1415926525
c T
    nr=(2*m+n)/ndr
    ns=-(2*n+m)/ndr
    nt2=3*l2/ndr/ndr
    nt=int(sqrt(dfloat(nt2))+eps)
c N
    nn=2*l2/ndr
c R
    ichk=0
    if(nr.eq.0) then
        n60=1
    else
        n60=nr
    endif
c
    itest2=1
c
    do 60 np=-abs(n60),abs(n60)
    do 61 nq=-abs(ns),abs(ns)
    j2 = nr*nq - ns*np
    if(j2.eq.1) then
    j1 = m *np - n*nq
    if( j1.gt.0 .and. j1.lt.nn ) then
        ichk=ichk+1
        nnp(ichk)=np
        nnq(ichk)=nq
    endif
    endif
61 continue
60 continue
c
    if(ichk.eq.0) then
        stop ' not found p,q strange!!'
    endif
c
```

```
        itest3=1
c
        if(ichk.ge.2) then
        if(itest3.eq.1) then
        stop ' more than 1 pair of p,q strange!!'
        endif

        if(nr.ne.0 .and. ns.ne.0) then
c
        if(itest1.eq.1) then
        do 77 i=1,ichk
        if((m*nnp(i)-n*nnq(i)).lt.nn) goto 777
77       continue
        endif
c
        endif
        endif
777      continue
c
        if(itest.eq.1) then
        np=nnp(1)
        nq=nnq(1)

        endif
2       continue
        return
1       continue
        stop

        end

c
c calculate the highest common divisor
c
        integer function igcm(ii,jj)
        i=abs(ii)
        j=abs(jj)
        if(j.gt.i) then
        iw=j
        j=i
        i=iw
        endif
        if(j.eq.0) then
        igcm=i
        return
        endif
10      continue
        ir=mod(i,j)
        if(ir.eq.0) then
        igcm=j
        return
        else
        i=j
        j=ir
        endif
        goto 10
        end
```

B.2 nT ナノチューブの座標を求めるプログラム

1. 場所:

file= takeya/for/tube/nanbai.f

2. 実行方法:

(1) tube-xyz.f を実行

(2) 次に nanbai.f を実行する。

display 上に ' チューブを何倍する? n=' と出てくるので、display 上に N 倍したいときは、N

と入力すれば良い。3. 入力 file=en.xyz2(tube-xyz.f を実行することによってできる。)

出力 file=tube4.xyz(Nt チューブの xmol 用座標)

file=tube.near(Nt チューブの最近接データ)

```

c
c   file=~takeya/for/tube/nanbai.f
c
c   nT の program
c   programu by T.Takeya
c
c
c   implicit real*8(a-h,o-z)
c   parameter (nk=5000)
c   dimension x(nk),y(nk),z(nk)
c   dimension iic(nk),ic(nk,3),iz(nk,3)
c
c   write(*,*) ' チューブを何倍する? n='
c   read(*,*)n
c   open(61,FILE='en.xyz2')
c   read(61,*)nn
c   read(61,*)t
c   do 10 i=1,nn
c       read(61,*)j,x(i),y(i),z(i)
10      continue
c       close(61)
c       write(*,*) ' チューブの長さ L=',n*t
c       aa=1.42d0
c       open(60,FILE='en.xyz2')
c       write(60,*)nn*n
c       write(60,124)t*n,aa
c           do k=1,n
c       do 11 i=1,nn
c           write(60,123)i*k,x(i),y(i),z(i)+t*k
124      format(f20.10,' ',f20.10)
123      format(i5,' ',f20.10,' ',f20.10,' ',f20.10)
11      continue
c           end do
c       close(60)
c       write(*,*) ' チューブの原子数 =',n*nn
c       ii=nn
c       do 20 i=1,nn
c           do 21 jj=0,n-1
c               iii=ii+i+jj*ii
c               z(iii)=z(i)+(jj+1)*t
c               x(iii)=x(i)
c               y(iii)=y(i)
21      continue
20      continue
c

```

```
c      Nt チューブの座標の出力 (xmol deta)
c
      open(60,FILE='Ntube.xyz')
      write(60,*)n*nn
      write(60,*)' '
      do 100 i=1,n*nn
          write(60,700)x(i),y(i),z(i)
100      continue
700      format('C',3f10.5)
      close(60)

c
c      初期化
c
      do i=1,n*nn
          iic(i)=0
          do j=1,3
              ic(i,j)=0
              iz(i,j)=0
          end do
      end do

      do i=1,n*nn
          k=0
          do j=1,n*nn
              bx=x(i)-x(j)
              by=y(i)-y(j)
              bz=z(i)-z(j)
              bb=sqrt(bx*bx+by*by+bz*bz)
              if((bb.gt.0.3d0).and.(bb.lt.1.6d0)) then
                  k=k+1
                  iic(i)=i
                  ic(i,k)=j
                  iz(i,k)=1
              endif
          end do
      end do

c
c
c
      open(60,FILE='tube.near')
      do i=1,n*nn
          write(60,16)iic(i),(ic(i,i1),i1=1,3),(iz(i,i2),i2=1,3)
16      format(7i5)
      end do
      close(60)

      open(60,FILE='saikn-n')
      write(60,*)n*nn
      write(60,*)n*t
      i4=0
      do 900 i=1,n*nn
          ik=i4+i
          write(60,750)ik,x(i),y(i),z(i)
750      format(' ',i4,3f10.5)
900      continue
      close(60)

c
```

```
stop  
end
```



```
      call Khani(nj,nk,rkx,rky)
c      write(*,*)k+1
c
c
c      位相係数を求める。
c
c
c
c      do 20 i=1,nf
c      if(i.eq.1) then
c          nnn=int(float(nj)*sqrt(3.0)/2.0)
c          endif
c
c      if(i.eq.2) then
c          nnn=int(float(nj)/2.0)
c          endif
c
c      if(i.eq.3) then
c          nnn=nj+1
c          endif
c
c
c      do 30 j=1,nnn
c          do 40 jj=1,nk
c      otA(jj)=exp(oi*(x(jj)*rkx(i,j)+y(jj)*rky(i,j)))
c      otB(jj)=exp(-oi*(x(jj)*rkx(i,j)+y(jj)*rky(i,j)))
40      continue
c
c
c
c      D の matrix 成分 (DAA,DAB,DBB,DBA) を求める subroutine(Dmatrix) を呼ぶ。
c
c
c
c      call Dmatrix(nk,otA,otB,a,b,oDAA,oDBB,oDAB,oDBA,n,n1)
c
c
c
c      D を求める。
c
c
c      call Ddou(oDAA,oDAB,oDBB,oDBA,oD,n,n1)
c
c      pm; 原子の質量
c      pm=12.0d0/(6.022137d0)
c      cc=2.99792458d0
c      do ipp=1,n
c          do ip=1,n1
c              oD(ip,ipp)=oD(ip,ipp)/pm*0.1d0
c          end do
c      end do
c
c      oD を対角化する。
c      行列を対角化する subroutine(deigch) を呼ぶ。
c
c
c      call deigch(oD,n,n1,ne,nv,eps,w,lw,e,V)
```

```
c
c   個有値をそれぞれに当てはめる
c
c
c
c
c       nj1=int(float(nj)*sqrt(3.0)/2.0)
c       nj2=int(float(nj)/2.0)
c
c   do 50 ik=1,n
c       if(e(ik).lt.0.0) then
c           write(*,*)'- j',j,e(ik)
c           e(ik)=e(ik)*0.0
c           endif
c       if(i.eq.1) then
c       eee(ik,j)=sqrt(e(ik))
c       endif
c
c       if(i.eq.2) then
c           eee(ik,j+nj1)=sqrt(e(ik))
c       endif
c
c       if(i.eq.3) then
c           eee(ik,j+nj1+nj2)=sqrt(e(ik))
c           if(j.eq.3) then
c               ek(ik)=eee(ik,j)/cc/2.0d0/pi*10000d0
c               write(*,*)ek(ik)
c           endif
c       endif
c
c
c
c   点の個有値、個有ベクトルを求める
c
c
c       if((i.eq.3).and.(j.eq.100)) then
c           kko=j+nj1+nj2
c           ek(ik)=eee(ik,kko)/cc/2.0d0/pi*10000d0
c           k=j
c           do jkf=1,n
c               vk(jkf,ik)=v(jkf,ik)
c           end do
c       endif
c
c
c   50   continue
c   30   continue
c   20   continue
c       write(*,*)(ek(k),k=1,n),k
c
c   出力ファイル
c
c
c       mnm=236
c       nll=100
c       st=pi/6.0d0/6.0d0*0.0d0
c       ddk=aa
c       write(*,*)nj1,nj2,nj
c       rm=5.5937692037053D-06
c       write(*,*) rm/cc/2.0d0/pi*1d1
c       &*2.0d0*cc*ddk*nll*pi/3.6275987d0
```

```

        open(60,file='gn-phonon')
        iii=0
c
        do 60 i=1,n/2
            iii=iii+1
            do 70 jk=1,nj+nj1+nj2+1
                rk=float(jk)
                write(60,200) rk,eee(iii,jk)/cc/2.0d0/pi*10000d0
                if(jk.eq.mnm) then
                    write(*,*) rk,(eee(iii,jk))/cc/2.0d0/pi*1d1
&*2.0d0*cc*ddk*nll*pi/3.6275987d0
                endif
70            continue
            iii=iii+1
            do 80 jk=nj+nj1+nj2+1,1,-1
                rk=float(jk)
                write(60,200) rk,eee(iii,jk)/cc/2.0/pi*10000d0
                if(jk.eq.mnm) then
                    write(*,*) rk,eee(iii,jk)/cc/2.0d0/pi*1d1
&*2.0d0*cc*ddk*nll*pi/3.6275987d0
                endif
80            continue
60            continue
200            format(E10.5,' ',E10.5)
                close(60)
c
c            点 の個有値 (file name=g-eval) の出力
c
        open(60,file='g-eval')
            write(60,300) (ek(j),j=1,n)
            write(60,*) ' '
            do 320 k=1,n
                write(60,300) (vk(k,j),j=1,n)
300            format(100f10.4)
320            continue
            close(60)
c
c            点 の個有ベクトル (file name=g-evec) の出力
c
        open(60,file='g-evec')
            do 500 i=1,n
                do 600 j=1,n
                    write(60,310) i,j,v(i,j)
310            format('evector(',i5,',',i5,')=',f10.5,f10.5)
600            continue
500            continue
            close(60)
c
c
c
        open(60,file='jm')
        write(60,*)n
        close(60)
c
c
c
        stop

```

```

end
-----c
c
c   D を求める subroutine(Ddou)
c
-----c
      subroutine Ddou(oDAA,oDAB,oDBB,oDBA,oD,n,n1)
      implicit real*8(a-h,p-y)
      implicit complex*16(o)
      dimension oDAA(n/2,n1/2),oDBB(n/2,n1/2)
      dimension oDAB(n/2,n1/2),oDBA(n/2,n1/2),oD(n,n1)
c
c
c   D の初期化
c
      do 10 i=1,n
      do 20 ii=1,n1
      oD(i,ii)=(0.0d0,0.0d0)
20    continue
10    continue
c
c
      do 40 i=1,n
      do 50 j=1,n1
      if(i.le.n/2) then
      if(j.le.n1/2) then
      oD(i,j)=oDAA(i,j)
      else
      oD(i,j)=oDAB(i,j-n1/2)
      endif
      endif
      if(i.ge.n/2+1) then
      if(j.le.n1/2) then
      oD(i,j)=oDBA(i-n/2,j)
      else
      oD(i,j)=oDBB(i-n/2,j-n1/2)
      endif
      endif
50    continue
40    continue
      return
      end
-----c
c
c   K の範囲を定める subroutine(Khani)
c
-----c
      subroutine Khani(nj,nk,rkx,rky)
      implicit real*8(a-h,p-y)
      parameter (nf=3)
      dimension rkx(nf,nj+1),rky(nf,nj+1)
c
      pi=atan(1.0)*4.0
c a program for kx, ky data for graphite
c
c   K-G-M-K line   R. Saito Oct.28 1991
c

```

```
c
c 0.0  0.0      gamma
c 3.6275987 0.0      M
c 3.6275987 2.0943951 K
c
c 角 KGM=st
      st=pi/6.0d0/6.0d0*0.0d0
      ss=abs(sin(st))
      cs=abs(cos(st))
c
c
c
c from G to M
c
c
c
c
      x1=0.0d0
      x2=3.6275987d0
      y1=0.0d0
      y2=0.0d0
      n=int(sqrt(3.0)*float(nj)/2.0)
c
      dx=(x2-x1)/float(n)
      dy=(y2-y1)/float(n)
c
      do 1 i=1,n
          rkx(1,i)=x1+dx*float(i-1)
          rky(1,i)=0.0d0
1      continue
c
c from M to K
c
      x1= 3.6275987d0
      x2=3.6275987d0*cs
      y1=0.0d0
      y2=3.6275987d0*ss
c
cc      y2= -3.6275987d0*abs(tan(st))
      n=int(float(nj)/2.0)
c
      dx=(x2-x1)/float(n)
      dy=(y2-y1)/float(n)
c
      do 2 i=1,n
          rkx(2,i)=x1
          rky(2,i)=y1+dy*float(i-1)
2      continue
c
c from K to
c
      x1= 3.6275987d0*cs
      x2=0.0d0
      y1= 3.6275987d0*ss
      y2=0.0d0
      n=nj
      write(*,*) 'nj=',n
c
      dx=(x1-x2)/float(n)
```

```

dy=(y1-y2)/float(n)
c
do 3 i=1,n
  rkx(3,i)=x1-dx*float(i-1)
c
  rkx(3,i)=3.6275987d0*cs*float(i)
  rky(3,i)=3.6275987d0*ss/float(i)
3
  continue
  rkx(3,n+1)=0.0d0
  rky(3,n+1)=0.0d0
  return
end

-----c
c
c
c   D の成分 (DAA,DAB,DBB,DBA) を求める subruitn(Dmatrix)
c
c
-----c
  subroutine Dmatrix(nk,otA,otB,a,b,oDAA,oDBB,oDAB,oDBA,n,n1)
c
  implicit real*8(a-h,p-y)
  implicit complex*16(o)
  dimension otA(nk),otB(nk)
  dimension oDAA(n/2,n1/2),oDAB(n/2,n1/2)
  dimension oDBB(n/2,n1/2),oDBA(n/2,n1/2)
  dimension a(nk,n/2,n1/2),b(nk,n/2,n1/2)
c
c
  pi=atan(1.0d0)*4.0d0
c
c
c
  do 10 ii=1,n/2
  do 20 iii=1,n1/2
    oDAA(ii,iii)=(0.0d0,0.0d0)
    oDBB(ii,iii)=(0.0d0,0.0d0)
    oDAB(ii,iii)=(0.0d0,0.0d0)
    oDBA(ii,iii)=(0.0d0,0.0d0)
  20  continue
  10  continue
c
c
  do 30 i=1,nk
    do 40 j=1,n/2
      do 50 jj=1,n1/2
c
c   DAA,DBB を求める
c
  if((i.ge.4).and.(i.le.9)) then
c
    oDAA(j,jj)=oDAA(j,jj)-otA(i)*a(i,j,jj)+a(i,j,jj)
    oDBB(j,jj)=oDBB(j,jj)-otB(i)*b(i,j,jj)+b(i,j,jj)
  else
c

```

```

c   DAB,DBA を求める。
c
      oDAA(j,jj)=oDAA(j,jj)+a(i,j,jj)
      oDBB(j,jj)=oDBB(j,jj)+b(i,j,jj)
      oDAB(j,jj)=oDAB(j,jj)-otA(i)*a(i,j,jj)
      oDBA(j,jj)=oDBA(j,jj)-otB(i)*b(i,j,jj)
      endif
c
50   continue
40   continue
30   continue
      return
      end
c
c-----c
c o.k
c
c   KA と KB; 力の定数を求める subruitn(Kmatrix)
c
c-----c
      subroutine Kmatrix(nk,x,y,z,a,b,n,n1)
      implicit real*8(a-h,p-z)
      dimension x(nk),y(nk),z(nk)
      dimension a(nk,n/2,n1/2),b(nk,n/2,n1/2)
c
      pi=atan(1.0)*4.0
c
      do 10 i=1,nk
c
          if(i.le.3) then
              fr=36.5d0
              fti=24.5d0
              fto=9.82d0
              endif
c
          if((i.ge.4).and.(i.le.9)) then
              fr=8.80d0
              fti=-3.23d0
              fto=-0.40d0
              endif
c
          if((i.ge.10).and.(i.le.12)) then
              fr=3.00d0
              fti=-5.25d0
              fto=0.15d0
              endif
c
          if((i.ge.13).and.(i.le.18)) then
              fr=-1.92d0
              fti=2.29d0
              fto=-0.58d0
              endif
c
c   KA の成分を求める。(st,st+pi で行列成分は同じになる。)
c
c
c

```



```

        if(x(i).eq.0) then
            st=pi/2.0d0
        else
            st=atan(y(i)/x(i))
        endif
c
c     行列成分;a11,a12,a13,...a33
c
        cs=cos(st)
        ss=sin(st)
        a(i,1,1)=fr*cs*cs+fti*ss*ss
        a(i,1,2)=fr*cs*ss-fti*ss*cs
        a(i,1,3)=0.0d0
        a(i,2,1)=a(i,1,2)
        a(i,2,2)=fr*ss*ss+fti*cs*cs
        a(i,2,3)=0.0d0
        a(i,3,1)=0.0d0
        a(i,3,2)=0.0d0
        a(i,3,3)=fto
c
c
c     KBの成分を求める。
c
c
        b(i,1,1)=a(i,1,1)
        b(i,1,2)=a(i,1,2)
        b(i,1,3)=0.0d0
        b(i,2,1)=a(i,2,1)
        b(i,2,2)=a(i,2,2)
        b(i,2,3)=0.0d0
        b(i,3,1)=0.0d0
        b(i,3,2)=0.0d0
        b(i,3,3)=fto
c
c
10     continue
        return
        end
c
c-----c
c
c     行列を対角化する subroutine(deigch)
c
c-----c
c
SUBROUTINE DEIGCH(A,N,N1,NE,NV,EPS,W,LW,E,V)
IMPLICIT REAL*8 (A-H, O-Z)
COMPLEX*16 A,V,CR,CS,X
LOGICAL SW, LW
DIMENSION A(N1,N1), E(N1), V(N1,N1), W(N1,7), LW(N1)
DREAL(X)=X
DIMAG(X)=-X*(0.0D0,1.0D0)
X=X
IF(N.LE.0 .OR. NE.EQ.0 ) GO TO 910
NEA=IABS(NE)
NVA=IABS(NV)

```

```

      IF(N1.LT.N .OR. N.LT.NEA .OR. NEA.LT.NVA ) GO TO 920
      IF(EPS.LT.0.DO) EPS=1.D-16
      NM1=N-1
      N2=N-2
      IF(N2) 10, 20, 50
C   WHEN N=1
10  E(1)=A(1,1)
      IF( NV.NE.0 ) V(1,1) = 1.ODO
      GO TO 900
C   WHEN N=2
C   COMPUTE EIGENVALUES OF 2*2 MATRIX
20  CALL ERRSET(207,256,-1,1)
      W(1,1)=A(1,1)
      W(2,1)=A(2,2)
      W(1,2)=CDABS(A(2,1))
      A(1,1)=A(2,1)/W(1,2)
      T = 0.5D0*(W(1,1)+W(2,1))
      R=W(1,1)*W(2,1)-W(1,2)**2
      D=T*T-R
      Q=DABS(T)+DSQRT(D)
      IF(T.LT.0.) Q=-Q
      T=T*DFLOAT(NE)
      IF(T) 40, 30, 30
30  E(1)=Q
      IF(NEA.EQ.2) E(2)=R/Q
      GO TO 310
40  E(1)=R/Q
      IF(NEA.EQ.2) E(2)=Q
      GO TO 310
C   WHEN N=3,4,...
C   REDUCE TO TRIDIAGONAL FORM BY HOUSEHOLDER'S METHOD
50  DO 60 I=1,N
60  W(I,1)=A(I,I)
      DO 190 K=1,N2
      K1=K+1
      S=0.
      DO 70 I=K1,N
70  S=S+DREAL(A(I,K))**2+DIMAG(A(I,K))**2
      SR = DSQRT(S)
      T=CDABS(A(K1,K))
      W(K,2)=-SR
      IF(T) 90, 80, 90
80  A(K,K) = 1.ODO
      GO TO 100
90  A(K,K)=A(K1,K)/T
100 IF(S.EQ.0.) GO TO 190
      R = 1.ODO/(S+T*SR)
      A(K1,K)=A(K1,K)+SR*A(K,K)
      DO 140 I=K1,N
      CS=(0.,0.)
      IF(I.EQ.K1) GO TO 120
      IM1 = I-1
      DO 110 J=K1,IM1
110 CS=CS+A(I,J)*A(J,K)
120 CS=CS+W(I,1)*A(I,K)
      IF(I.EQ.N) GO TO 140
      IP1 = I+1

```

```

      DO 130 J=IP1,N
130 CS=CS+DCONJG(A(J,I))*A(J,K)
140 A(I,I)=CS*R
      CS=(0.,0.)
      DO 150 I=K1,N
150 CS=CS+DCONJG(A(I,K))*A(I,I)
      CS = 0.5DO*R*CS
      DO 160 I=K1,N
160 A(I,I)=A(I,I)-CS*A(I,K)
      DO 170 I=K1,N
170 W(I,1) = W(I,1)-2.ODO*DREAL(A(I,K)*DCONJG(A(I,I)))
      IP1 = K+2
      DO 180 I=IP1,N
      IM1 = I-1
      DO 180 J=K1,IM1
180 A(I,J)=A(I,J)-A(I,K)*DCONJG(A(J,J))-A(I,I)*DCONJG(A(J,K))
190 CONTINUE
      W(NM1,2)=CDABS(A(N,NM1))
      A(NM1,NM1)=A(N,NM1)/W(NM1,2)
C COMPUTE EIGENVALUES BY BISECTION METHOD
      CALL ERRSET(207,256,-1,1)
      R=DMAX1((DABS(W(1,1))+DABS(W(1,2))), (DABS(W(NM1,2))+DABS(W(N,1))))
      DO 200 I=2,NM1
      T=DABS(W(I-1,2))+DABS(W(I,1))+DABS(W(I,2))
      IF(T.GT.R) R=T
200 CONTINUE
      EPS1=R*0.1D-15
      EPS2=R*EPS
      DO 210 I=1,NM1
210 W(I,3)=W(I,2)**2
      IF(NE.LT.0) R=-R
      F=R
      DO 220 I=1,NEA
220 E(I)=-R
      DO 300 K=1,NEA
      D=E(K)
230 T = 0.5DO*(D+F)
      IF(DABS(D-T).LE.EPS2 .OR. DABS(F-T).LE.EPS2 ) GO TO 300
      J=0
      I=1
240 Q=W(I,1)-T
250 IF(Q.GE.0.) J=J+1
      IF(Q.EQ.0.) GO TO 260
      I=I+1
      IF(I.GT.N) GO TO 270
      CALL OVERFL(L)
      Q=W(I,1)-T-W(I-1,3)/Q
      CALL OVERFL(L)
      IF(L.NE.1) GO TO 250
      J=J+1
      I=I-1
260 I=I+2
      IF(I.LE.N) GO TO 240
270 IF(NE.LT.0) J=N-J
      IF(J.GE.K) GO TO 280
      F=T
      GO TO 230

```

```
280 D=T
    M=MINO(J,NEA)
    DO 290 I=K,M
290 E(I)=T
    GO TO 230
300 E(K)=T
C COMPUTE EIGENVECTORS BY INVERSE ITERATION
310 CALL ERRSET(207, 10, 5,2)
    IF(NV.EQ.0) GO TO 900
    MM=584287
    CALL ERRSET(207,256,-1,1)
    DO 490 I=1,NVA
    DO 320 J=1,N
        W(J,3)=W(J,1)-E(I)
        W(J,4)=W(J,2)
320 W(J,7) = 1.0D0
        SW=.FALSE.
C REDUCE TO TRIANGULAR FORM
    DO 340 J=1,NM1
        IF(DABS(W(J,3)).LT.DABS(W(J,2))) GO TO 330
        IF(W(J,3).EQ.0.) W(J,3)=1.0D-30
        W(J,6)=W(J,2)/W(J,3)
        LW(J)=.FALSE.
        W(J+1,3)=W(J+1,3)-W(J,6)*W(J,4)
        W(J,5)=0.
        GO TO 340
330 W(J,6)=W(J,3)/W(J,2)
        LW(J)=.TRUE.
        W(J,3)=W(J,2)
        T=W(J,4)
        W(J,4)=W(J+1,3)
        W(J,5)=W(J+1,4)
        W(J+1,3)=T-W(J,6)*W(J,4)
        W(J+1,4)=-W(J,6)*W(J,5)
340 CONTINUE
        IF(W(N,3).EQ.0.) W(N,3)=1.0D-30
C BEGIN BACK SUBSTITUTION
        IF(I.EQ.1 .OR. DABS(E(I)-E(I-1)).GE.EPS1) GO TO 360
C GENERATE RANDOM NUMBERS
    DO 350 J=1,N
        MM=MM*48828125
350 W(J,7)=FLOAT(MM)*0.4656613E-9
360 CALL OVERFL(L)
        T=W(N,7)
        R=W(N-1,7)
370 W(N,7)=T/W(N,3)
        W(N-1,7)=(R-W(N-1,4)*W(N,7))/W(N-1,3)
        CALL OVERFL(L)
        IF(L.NE.1) GO TO 390
        DO 380 J=1,N2
380 W(J,7)=W(J,7)*1.0D-5
        T=T*1.0D-5
        R=R*1.0D-5
        GO TO 370
390 IF(N.EQ.2) GO TO 440
        K=N2
400 T=W(K,7)
```

```
410 W(K,7)=(T-W(K,4)*W(K+1,7)-W(K,5)*W(K+2,7))/W(K,3)
    CALL OVERFL(L)
    IF(L.NE.1) GO TO 430
    DO 420 J=1,N
420 W(J,7)=W(J,7)*1.0D-5
    T=T*1.0D-5
    GO TO 410
430 K=K-1
    IF(K) 440,440,400
440 IF(SW) GO TO 470
    SW=.TRUE.
    DO 460 J=1,NM1
    IF(LW(J)) GO TO 450
    W(J+1,7)=W(J+1,7)-W(J,6)*W(J,7)
    GO TO 460
450 T=W(J,7)
    W(J,7)=W(J+1,7)
    W(J+1,7)=T-W(J,6)*W(J+1,7)
460 CONTINUE
    GO TO 360
470 DO 480 J=1,N
480 V(J,I)=W(J,7)
490 CONTINUE
C BEGIN BACK TRANSFORMATION (1)
    CR = 1.0D0
    DO 500 J=2,N
    CR=CR*A(J-1,J-1)
    DO 500 I=1,NVA
500 V(J,I)=V(J,I)*CR
C BEGIN BACK TRANSFORMATION (2)
    CALL ERRSET(207, 10, 5,2)
    IF(N.EQ.2) GO TO 600
    DO 590 I=1,NVA
    K=N2
550 CR=-A(K+1,K)*DCONJG(A(K,K))*W(K,2)
    IF(DREAL(CR).EQ.0.0 .AND. DIMAG(CR).EQ.0.0) GO TO 580
    CR = 1.0D0/CR
    CS=(0.,0.)
    IP1 = K+1
    DO 560 J=IP1,N
560 CS=CS+DCONJG(A(J,K))*V(J,I)
    CR=CR*CS
    DO 570 J=IP1,N
570 V(J,I)=V(J,I)-CR*A(J,K)
580 K=K-1
    IF(K.GE.1) GO TO 550
590 CONTINUE
600 CONTINUE
C NORMALIZE EIGENVECTORS
C NORMALIZE AS MAXIMUM ELEMENT=1
    DO 620 I=1,NVA
    T=DABS(DREAL(V(1,I)))+DABS(DIMAG(V(1,I)))
    K=1
    DO 610 J=2,N
    R=DABS(DREAL(V(J,I)))+DABS(DIMAG(V(J,I)))
    IF(T.GE.R) GO TO 610
    T=R
```

```
      K=J
610 CONTINUE
      CR = 1.0D0/V(K,I)
      DO 620 J=1,N
620 V(J,I)=V(J,I)*CR
      IF(NV.LT.0) GO TO 900
C ORTHOGONALIZE AS NORM=1
      DO 680 I=1,NVA
      IF(I.EQ.1 .OR. DABS(E(I)-E(I-1)).GT.EPS) GO TO 650
      IM1 = I-1
      DO 640 J=M,IM1
      CS=(0.,0.)
      DO 630 K=1,N
630 CS=CS+DCONJG(V(K,J))*V(K,I)
      DO 640 K=1,N
640 V(K,I)=V(K,I)-CS*V(K,J)
      GO TO 660
650 M=I
C NORMALIZE AS NORM=1
660 S=0.
      DO 670 J=1,N
670 S=S+DREAL(V(J,I))**2+DIMAG(V(J,I))**2
      T = DSQRT(1.0D0/S)
      DO 680 J=1,N
680 V(J,I)=V(J,I)*T
900 RETURN
C PRINT ERROR MESSAGE
910 WRITE(6,1000) N,NE
      GO TO 900
920 WRITE(6,1100) NV,NE,N,N1
      GO TO 900
1000 FORMAT(1H0,'(SUBR. DEIGCH) N=',I5,',NE=',I5,' N SHOULD BE GREATER
1 THAN ZERO AND NE SHOULD BE NON-ZERO. RETURN WITH NO CALCULATION.
2 ')
1100 FORMAT(1H0,'(SUBR. DEIGCH) NV=',I5,',NE=',I5,',N=',I5,',N1=',I5,
1 ' NV,NE,N,N1 SHOULD SATISFY THE FOLLOWING INEQUALITIES, !NV! <= !N
2E! <= N <= N1 .' /1H ,'RETURN WITH NO CALCULATION.' )
      END
      SUBROUTINE ERRSET(I,J,K,L)
      RETURN
      END
      SUBROUTINE OVERFL(L)
      RETURN
      END
```

B.4 ナノチューブ分散関係を求めるプログラム

```

場所: takeya/for/tube/tu-phonon1.f
入力: en.xyz2 (プログラム tube-xyz.f)
t-ch (プログラム tube-xyz.f)
出力: tu-phonon1.dat
: tu-eval. 固有値、個有ベクトル

%
%
%
c
c----- tube の phonon の分散関係を求めるプログラム -----c
c
c   file=~takeya/for/tube/tu-phonon1.f
c
c   programed by takao takeya
c
c   date 96.10.07
c       97.03.03 modified by R. Saito
c
c   入力 en.xyz2 (プログラム tube-xyz.f)
c       t-ch (プログラム tube-xyz.f)
c   出力 tu-phonon1.dat
c
c   nk; 原子数, ns; 最大近接原子数, nj; k 点の分割数
c   ndmax; 求められる最大近接 (第何 ndmax 近接まで) s
c   ndmax1; 第何近接まで求めるか? (1=<ndmax1=<ndmax)
c       implicit real*8(a-h,p-z)
c       implicit complex*16(o)
c
c   注; 原子数; nk を変える
c
c   parameter (nk=40,n1=nk*3,n=n1,ne=n1,nv=n1,nj=1,ns=18)
c   parameter (ndmax=4,ndmax1=4)
c
c   logical lw
c   complex*16 oD,V
c   dimension oD(n1,n),V(n1,n1),vk(n1,n1),ek(n),epo(n1)
c   dimension x(nk),y(nk),z(nk),ic(nk,ns),rz(nk,ns)
c   dimension a(nk,ns,3,3),iken(nk,ns),a2(nk,ns,3,3)
c   dimension qqq(nk),eee(nk*3,nj+1),xx(nk),qka(nk,ns)
c   dimension w(n1,7),e(n1),lw(n1),qkaa(nk,ns),dn(ndmax)
c   dimension fr(ndmax),fti(ndmax),fto(ndmax),nkc(nk,ns)
c   dimension a3(nk,ns,3,3),a4(nk,ns,3,3),daa(nk,ns)
c   dimension ar(4),ati(4),ato(4)
c
c   pi=atan(1.0d0)*4.0d0
c   eps=1.0d-16
c
c   ppp=6.6890
c   write(*,*)165*(ppp/6.78)**(-1)
c
c   pm: 炭素原子の質量
c   cc: 光速速度
c
c   pm=12.0d0/(6.022137d0)
c   cc=2.99792458d0

```

```
do i=1,n1
  epo(i)=0.0d0
end do

c
c
c  nss; 第 ndmax1 までの近接原子の数を求める
      nss=0
      enk=0.0d0
      do i=1,ndmax1
        if(mod(i,2).eq.1) then
          nss=3+nss
        else
          nss=6+nss
        endif
      end do
      write(*,*) ' 近接原子数 =',nss

c
c  factor(0<factor=<1.0): k の最大値に対してどれぐらいまで求める
c                    か?
c                    =1.0 T/pi まで
      factor = 1.00d0

c
c  nn2; 原子数
c
      open(61,FILE='en.xyz2')
      read(61,*) nn2
      read(61,*) t,acc

c
c  write(*,*) ' カイラルベクトル C_h = (', nCh,',', mCh,')'
      if(nn2.ne.nk) stop 'nn2.ne.nk=change nk'
      do 10 i=1,nn2
        read(61,*) idum,x(i),y(i),z(i)
10      continue
      write(*,*)
      close(61)

c
c  aa=1.42d0

c
c  if(abs(aa-acc).gt.0.001) stop 'check acc or aa'

c
c  ch;|C_h|,t;|T|,qa;A 原子の周期,A 原子に対してのB 原子のずれ
c
      open(61,FILE='t-ch')
      read(61,21) t,ch,qa,qb,lo
21      format(5f25.20)
      close(61)

c
c  第四近接までの原子を求める subroutine を近接を呼ぶ。
c
      call kinsetu(nn2,ndmax,ndmax1,x,y,z,nk,ns
&,t,ic,rz,iken,ch,qa,qb,qqq,xx,qka,qkaa,dn,nss,nakc
&,daa)

c
c
c  力の定数を求める subroutine(Kmatrix) を呼ぶ。
c
c
```



```

      call Kmatrix(nn2,nk,ns,ndmax,ndmax1,a,a2,t,ch,qa,qb,
&iken,ic,qqq,rz,x,y,z,qka,qkaa,dn,nss,fr,fti,fto,nakc,a3,a4
&,daa,ar,ato,ati)
c
c      dkmax: k の最大値 zone は pi/t
c      nj: 分割数 (parameter 文で指定する。)
c      factor: 最大値に対してどれぐらいまで求めるか?
c
      kkk=0
      dkmax=pi/t*factor
      dk=dkmax/dfloat(nj)
      dk=49.19024d0*1.0
      write(*,*)1.0d0/dk*t/pi
c      dk=d0
c
c Main do loop: 分散関係を計算する。
c
      do 20 i=1,nj+1
          rt=1.0d0/dk*dfloat(i-1)
c
c      k 点に対する Dynamical Matrix を oD を求める。 subroutine(Dmatri
c
      call Dmatrix(nn2,oD,a,nk,ns,nss,n,n1,rt,rz,ic,a3,a4,a2)
c
c      oD を対角化する。
c      行列を対角化する subroutine(deigch) を呼ぶ。
c
c      固有ベクトルは 点 (i=1) の時にしか求めない。
c
cc      if(i.eq.1) then
cc          nnv=n
cc      else
cc          nnv=0
cc      end if
c
      nnv=n
      call deigch(oD,n,n1,ne,nnv,eps,w,lw,e,v)
c
c      個有値をそれぞれに当てはめる
c
      ekk1=0.0d0
      k=0
      do 30 j=1,n
          if(i.eq.1) then
              if(e(j).lt.eps)then
write(*,*) 'e(j)<0 , i , j = ',e(j),i,j
e(j)=0.0d0
endif
cc      endif
c
c
c
c
      k=k+1
30      continue
      do j=1,n
          eee(j,i)=sqrt(e(j))/cc/2.0d0/pi*1.0d3

```

```
        end do
c      点の個有値、個有ベクトルを求める。
c
        ng=1

        if(i.eq.ng) then
          do j=1,n
            ek(j)=eee(j,i)
            do jj=1,n
              vk(jj,j)=v(jj,j)
            end do
          end do
        endif

cc
121      continue
c
c   k 点の main loop 終了
c
20      continue

c
c   音速を求める。(k/pi のとき)
c
c
        open(60,file='velo')
        do i=n1-50,n1
          write(60,111)i,
&eee(i,nj+1)*cc/0.0001*t*2.0/10d2
111      format(i5,' ',f10.5)
        end do
        close (60)

c
        write(*,*) eee(n1,nj+1)
        write(*,*) eee(n1-2,nj+1)
        write(*,*) eee(n1-3,nj+1)

        write(*,*) eee(n1,nj+1)*cc/0.001*t*2.0
        write(*,*) eee(n1-2,nj+1)*cc/0.001*t*2.0
        write(*,*) eee(n1-3,nj+1)*cc/0.001*t*2.0

c
c   出力 file(=tu-phonon1.dat) を作る。
c
        open(60,file='tu-phonon1.dat')

c
        k1=0
        do 40 j=1,nn2*3

c
c   j が 奇数
c
        if (mod(j,2).eq.1) then
          do 50 i=1,nj+1
            k1=k1+1
            rt=dk*dfloat(i-1)*t/pi
            write(60,100)rt,eee(j,i)
100      format(f10.5,' ',f12.5)
          50      continue
        end if
c
```

```
c j が 偶数
c
  else
    do 70 i=nj+1,1,-1
      kl=k1+1
      rt= dk*dfloat(i-1)*t/pi
      write(60,100)rt,eee(j,i)
70    continue
  endif
c
40  continue
c
      close(60)
      write(*,*)kl
c
c  点の固有値, 個有ベクトルの出力 file(tu-eval)
c  for xmol & raman 用
c
cc  kp=0
  open(60,file='tu-eval')
  nn=int(nn2*3/8)+1
c
  kpp=0
  do jji=1,nn
c    if((jji-1)*8+8.ge.nn2*3)
      if(kpp.eq.1) goto 210
      if((jji-1)*8+8.ge.nn2*3) then
        kpp=kpp+1
        write(60,200)(ek(i),i=1+8*(jji-1),nn2*3)
      else
        write(60,200)(ek(i),i=1+8*(jji-1),8+(jji-1)*8)
      endif
c
  write(60,*)' '
    do i=1,nn2*3
      if((jji-1)*8+8.ge.nn2*3) then
        write(60,200) (vk(i,j),j=1+8*(jji-1),nn2*3)
      else
        write(60,200) (vk(i,j),j=1+8*(jji-1),8+(jji-1)*8)
      endif
200    format(10000f10.4)
      end do
      write(60,*)' '
      end do
210    continue
  close(60)
c  check
c
c  以下はデータ解析用 解析をしたい時は、itest = 1 にする。
c
c  nvk : 見たい固有ベクトルの番号 (1-n) n がエネルギー最少のモード
c
c  出力 file g-eval : 固有値の値 (全部)
c      tasi : nvk 番目の固有ベクトル
c      itest=1
c
c
```

```

        nvk =116
c
        if(itest.eq.1) then
c
            open(60,file='g-eval')
            do i=1,n
                write(60,108)i, ek(i)
108      format(i4,2x,f10.3)
            end do
            close(60)
            open(60,file='tasi')
            do i=1,nn2
                write(60,220) i,x(i),y(i),z(i),
&      vk(3*i-2,nvk),vk(3*i-1,nvk),vk(3*i,nvk)
220      format(i5,6f15.10)
            end do
            close(60)
c
        end if
c
        stop
        end
c-----c
c
c
c      第四近接までの原子を求める subroutine(kinsetu)
c
c-----c
        subroutine kinsetu(nn2,ndmax,ndmax1,x,y,z,nk,ns,t,ic,rz,
&iken,ch,qa,qb,qqq,xx,qka,qkaa,dn,nss,nakc,daa)
        implicit real*8(a-h,p-z)
        implicit complex*16(o)
        dimension x(nk),y(nk),z(nk),rz(nk,ns),qqq(nk),xx(nk),qka(nk,ns)
        dimension ic(nk,ns),iken(nk,ns),qkaa(nk,ns),dn(ndmax),nakc(nk,ns)
        dimension daa(nk,ns)
c
c
c
        pi=atan(1.0d0)*4.0d0
        aa=1.42d0
        eps2=1.0d-6
c      第n(n=1,2,3,4) 近接までの距離 dn(n=1,2,3,4)
c
        dn(1)=aa
        dn(2)=dsqrt(3.0d0)*aa
        dn(3)=2.0d0*aa
        dn(4)=dsqrt(7.0d0)*aa
c
c      unit cell を何倍移動させたらいいか;nan*t
c
        nan=int(dn(ndmax1)/t)+1
c
c      tube の座標より原子の角度を求める。
c
        do i=1,nk
            if(abs(x(i)).le.eps2) then

```

```
        if(y(i).gt.-eps2) then
        qqq(i)=pi/2.0d0
        goto 18
        else
        qqq(i)=pi*3.0d0/2.0d0
        goto 18
        endif
    endif
c
c
        if(abs(y(i)).le.eps2) then
            if(x(i).gt.-eps2) then
                qqq(i)=0.0d0
                goto 18
            else
                qqq(i)=pi
                goto 18
            endif
        endif
c
c
        qqq(i)=atan(y(i)/x(i))
c
        if(qqq(i).le.eps2) then
        if(x(i).ge.0.0d0) then
        qqq(i)=qqq(i)+2.0d0*pi
        goto 18
        else
        qqq(i)=qqq(i)+pi
        goto 18
        endif
        endif
c
        if(qqq(i).gt.eps2) then
        if(x(i).gt.0.0d0) then
        qqq(i)=qqq(i)
        goto 18
        else
        qqq(i)=qqq(i)+pi
        goto 18
        endif
        endif
c
18    continue
    if(qqq(i).gt.2.0d0*pi)stop'qqq'
c    write(*,*)qqq(i)*180/pi
    end do
    rr=ch/(2.0d0*pi)
    open(60,file='kei')
    write(60,*)nk
    write(60,*)' '
    do i=1,nk
        write(60,12)rr*cos(qqq(i)),rr*sin(qqq(i)),z(i)
12    format('C',3f10.5)
    end do
    close(60)
c
```

```

c   rr;tube の半径
c
c       rr=ch/(2.0d0*pi)
do i=1,nn2
  xx(i)=rr*qqq(i)
end do
c       write(*,*)2.0*pi*rr,ch
c       write(*,*)qqq(2),xx(2)
c
c
c
c       第 n 近接までの最近接原子を求める。
c
c
c       nan2=int(dn(ndmax1)/ch)+4
c
c       do 10 i=1,nn2
c         k=0
c
c       ii:T 方向の移動
c
c       do 60 ii=1,nan
c         do 30 j=1,nn2
c
c           st=0.0d0
c           if(ii.eq.1) then
c
c       iii:Ch 方向への移動
c
c       do 123 iii=-nan2,nan2
c         x1=xx(j)+dfloat(iii)*ch-xx(i)
c         b1=sqrt( x1*x1 + ( z(j)-z(i) ) * ( z(j)-z(i) ) )
c
c       ユニットセル内の近接原子 (1 ~ ndmax1) を求める。
c
c
c       do ij=1,ndmax1
c         dmax=dn(ij)+0.1d0
c         if(ij.eq.1) then
c           dmin=0.3d0
c         else
c           dmin=dn(ij-1)+0.1d0
c         endif
c
c       if((b1.gt.dmin).and.(b1.lt.dmax)) then
c         k=k+1
c         ic(i,k)=j
c         rz(i,k)=z(j)-z(i)
c         iken(i,k)=ij
c         qka(i,k)=x1
c         qkaa(i,k)=xx(j)+dfloat(iii)*ch-xx(i)
c         nakc(i,k)=iii
c         daa(i,k)=sqrt(rz(i,k)*rz(i,k)+(x(j)-x(i))**2
&+(y(j)-y(i))**2)
c         endif
c       end do

```

```
123         continue
           endif
30         continue
c
c
c ユニットセル外にある近接原子;(+t)
c
           do 40 j=1,nn2
           do 124 iii=-nan2,nan2
           x1=xx(j)+dfloat(iii)*ch-xx(i)
           z2=z(j)+dfloat(ii)*t
           b2=dsqrt(x1*x1+(z2-z(i))*(z2-z(i)))
c
           do ij=1,ndmax1
           dmax=dn(ij)+0.1d0
           if(ij.eq.1) then
           dmin=0.3d0
           else
           dmin=dn(ij-1)+0.1d0
           endif
c
           if((b2.gt.dmin).and.(b2.lt.dmax)) then
           k=k+1
           ic(i,k)=j
           rz(i,k)=z2-z(i)
           iken(i,k)=ij
           qka(i,k)=x1
           qkaa(i,k)=xx(j)+dfloat(iii)*ch-xx(i)
           nakc(i,k)=iii
           daa(i,k)=dsqrt(rz(i,k)*rz(i,k)+(x(j)-x(i))**2
           &+(y(j)-y(i))**2)
           endif
           end do
124         continue
40         continue
c
c 外;(-t)
c
           do 50 j=1,nn2
           do 125 iii=-nan2,nan2
           x1=xx(j)+dfloat(iii)*ch-xx(i)
           z3=z(j)-dfloat(ii)*t
           b3=sqrt(x1*x1+(z3-z(i))*(z3-z(i)))
c
           do ij=1,ndmax1
           dmax=dn(ij)+0.1d0
           if(ij.eq.1) then
           dmin=0.3d0
           else
           dmin=dn(ij-1)+0.1d0
           endif
c
           if((b3.gt.dmin).and.(b3.lt.dmax)) then
           k=k+1
           ic(i,k)=j
           rz(i,k)=z3-z(i)
           iken(i,k)=ij
```

```

        qka(i,k)=x1
        qkaa(i,k)=xx(j)+dfloat(iii)*ch-xx(i)
        nakc(i,k)=iii
        daa(i,k)=dsqrt(rz(i,k)*rz(i,k)+(x(j)-x(i))**2
&+(y(j)-y(i))**2)
        endif
    end do
125     continue
50     continue
60     continue
        if(k.ne.nss) then
            write(*,*) 'nss,i,k <> =',nss ,i,k
        endif
        if(k.ne.nss) stop'k-kinsetu'
10    continue
        return
        end
c-----c
c
c
c    D の成分を求める subruitn(Dmatrix)
c
c-----c
        subroutine Dmatrix(nn2,oD,a,nk,ns,nss,n,n1,rt,rz,ic,a3,a4,a2)
        implicit real*8(a-h,p-z)
        implicit complex*16(o)
        dimension oD(n1,n),a3(nk,ns,3,3),a4(nk,ns,3,3)
        dimension a(nk,ns,3,3),rz(nk,ns),ic(nk,ns),a2(nk,ns,3,3)
        data oi/(0.0d0,1.0d0)/
c
c
c    pi=4.0*atan(1.0)
c
c    D の初期化
c
        do 1 i=1,n1
            do 2 j=1,n
                oD(i,j)=(0.0d0,0.0d0)
            2     continue
        1     continue
c
c
c    pm; 質量,cc; 光速
        pm=12.0d0/(6.022137d0)
        cc=2.99792458d0
c
        do 100 i=1,nn2
            k=0
            do 110 ii=1,nss
                kp=ic(i,ii)
                do 5 j=1,3
                    do 6 jj=1,3
c
                        kj=j+(i-1)*3
                        kjj=jj+(kp-1)*3
                        al=a(i,ii,j,jj)
                        oD(kj,kjj)=-exp(oi*rt*rz(i,ii))*al/(pm*0.1d0)

```



```

      & +oD(kj,kjj)
6      continue
5      continue
110   continue
100   continue
c
      do 10 i=1,nn2
          k=0
          do 20 ii=1,nss
              do j=1,3
                  do jj=1,3
                      k=k+1
                      al=a(i,ii,j,jj)
                      oD(j+(i-1)*3,jj+(i-1)*3)=oD(j+(i-1)*3,jj+(i-1)*3)
&                      +al/(pm*0.1d0)
                          end do
                      end do
20      continue
          if(k.ne.nss*9)stop'hen-Dmatrix'
10     continue
          return
          end
c-----*abs(cos(st1))-----c
c
c     KA と KB; 力の定数を求める subruitn(Kmatrix)
c
c
c-----c
      subroutine Kmatrix(nn2,nk,ns,ndmax,ndmax1,a,a2,t,ch,qa,qb,
& iken,ic,qqq,rz,x,y,z,qka,qkaa,dn,nss,fr,fti,fto,nakc,a3,a4
& ,daa,ar,ato,ati)
      implicit real*8(a-h,p-z)
      implicit complex*16(o)
      dimension a(nk,ns,3,3),a2(nk,ns,3,3),rz(nk,ns)
      dimension qqq(nk),iken(nk,ns),ic(nk,ns),qka(nk,ns)
      dimension x(nk),y(nk),z(nk),qkaa(nk,ns),dn(ndmax)
      dimension fr(ndmax),fti(ndmax),fto(ndmax),nakc(nk,ns)
      dimension a3(nk,ns,3,3),a4(nk,ns,3,3),daa(nk,ns)
      dimension ar(4),ati(4),ato(4)
c
      aa=1.42d0
      pi=atan(1.0d0)*4.0d0
      eps2=1.0d-6
      rr=ch/(2.0*pi)
c
c
c     第 n(n=1,ndmax) までの力の定数を設定
c     fr;Radial, fti,fto;Tangential
          fr(1)=36.5d0
          fti(1)=24.5d0
          fto(1)=9.82d0
c
          fr(2)=8.80d0
          fti(2)=-3.23d0
          fto(2)=-0.400d0
c
          fr(3)=3.00d0

```



```

      &*(1.0d0-abs(cos(st1/2.0d0)))
c      fto1=fti(ijj)*abs(sin(st2))/( abs(cos(st1/2.0d0)))
      fti1=fr(ijj)+
      &abs(cos(st2))*fr(ijj)*
      &(1.0d0-abs(cos(st1/2.0d0)))
c      fto1=fti(ijj)
c      fti1=fr(ijj)
c      fr1=fro(ijj)
      endif
      end do

c
      cs=cos(st2)
      ss=sin(st2)

c
      a2(i,ii,1,1)=fr1
      a2(i,ii,1,2)= 0.0d0
      a2(i,ii,1,3)= 0.0d0
      a2(i,ii,2,1)= 0.0d0
      a2(i,ii,2,2)=fti1* cs * cs + fto1 * ss * ss
      a2(i,ii,2,3)=( fti1 - fto1 ) * ss * cs
      a2(i,ii,3,1)=0.0d0
      a2(i,ii,3,2)=a2(i,ii,2,3)
      a2(i,ii,3,3)=fto1 * cs * cs + fti1 * ss * ss

c
c
      a11=a2(i,ii,1,1)
      a22=a2(i,ii,2,2)
      a23=a2(i,ii,2,3)
      a33=a2(i,ii,3,3)

c
      cs=cos(st+st1/2.0d0)
      ss=sin(st+st1/2.0d0)

c
      a(i,ii,1,1)= a11*cs*cs + a22*ss*ss
      a(i,ii,1,2)= (a11-a22)*cs*ss
      a(i,ii,1,3)=-a23*ss
      a(i,ii,2,1)= a(i,ii,1,2)
      a(i,ii,2,2)= a22*cs*cs + a11*ss*ss
      a(i,ii,2,3)= a23*cs
      a(i,ii,3,1)= a(i,ii,1,3)
      a(i,ii,3,2)= a(i,ii,2,3)
      a(i,ii,3,3)= a33
30      continue
20      continue

c
c
      return
      end
c-----c
c
c      行列を対角化する subroutine(deigch)
c
c-----c
c
SUBROUTINE DEIGCH(A,N,N1,NE,NV,EPS,W,LW,E,V)
IMPLICIT REAL*8 (A-H, O-Z)

```

```

COMPLEX*16 A,V,CR,CS,X
LOGICAL SW, LW
DIMENSION A(N1,N), E(N1), V(N1,N), W(N1,7), LW(N1)
DREAL(X)=X
DIMAG(X)=-X*(0.0D0,1.0D0)
X=X
IF(N.LE.0 .OR. NE.EQ.0 ) GO TO 910
NEA=IABS(NE)
NVA=IABS(NV)
IF(N1.LT.N .OR. N.LT.NEA .OR. NEA.LT.NVA ) GO TO 920
IF(EPS.LT.0.D0) EPS=1.D-16
NM1=N-1
N2=N-2
IF(N2) 10, 20, 50
C WHEN N=1
10 E(1)=A(1,1)
   IF( NV.NE.0 ) V(1,1) = 1.0D0
   GO TO 900
C WHEN N=2
C COMPUTE EIGENVALUES OF 2*2 MATRIX
20 CALL ERRSET(207,256,-1,1)
   W(1,1)=A(1,1)
   W(2,1)=A(2,2)
   W(1,2)=CDABS(A(2,1))
   A(1,1)=A(2,1)/W(1,2)
   T = 0.5D0*(W(1,1)+W(2,1))
   R=W(1,1)*W(2,1)-W(1,2)**2
   D=T*T-R
   Q=DABS(T)+DSQRT(D)
   IF(T.LT.0.) Q=-Q
   T=T*DFLOAT(NE)
   IF(T) 40, 30, 30
30 E(1)=Q
   IF(NEA.EQ.2) E(2)=R/Q
   GO TO 310
40 E(1)=R/Q
   IF(NEA.EQ.2) E(2)=Q
   GO TO 310
C WHEN N=3,4,...
C REDUCE TO TRIDIAGONAL FORM BY HOUSEHOLDER'S METHOD
50 DO 60 I=1,N
60 W(I,1)=A(I,I)
   DO 190 K=1,N2
   K1=K+1
   S=0.
   DO 70 I=K1,N
70 S=S+DREAL(A(I,K))**2+DIMAG(A(I,K))**2
   SR = DSQRT(S)
   T=CDABS(A(K1,K))
   W(K,2)=-SR
   IF(T) 90, 80, 90
80 A(K,K) = 1.0D0
   GO TO 100
90 A(K,K)=A(K1,K)/T
100 IF(S.EQ.0.) GO TO 190
   R = 1.0D0/(S+T*SR)
   A(K1,K)=A(K1,K)+SR*A(K,K)

```

```
DO 140 I=K1,N
  CS=(0.,0.)
  IF(I.EQ.K1) GO TO 120
  IM1 = I-1
  DO 110 J=K1,IM1
110 CS=CS+A(I,J)*A(J,K)
120 CS=CS+W(I,1)*A(I,K)
  IF(I.EQ.N) GO TO 140
  IP1 = I+1
  DO 130 J=IP1,N
130 CS=CS+DCONJG(A(J,I))*A(J,K)
140 A(I,I)=CS*R
  CS=(0.,0.)
  DO 150 I=K1,N
150 CS=CS+DCONJG(A(I,K))*A(I,I)
  CS = 0.5D0*R*CS
  DO 160 I=K1,N
160 A(I,I)=A(I,I)-CS*A(I,K)
  DO 170 I=K1,N
170 W(I,1) = W(I,1)-2.0D0*DREAL(A(I,K)*DCONJG(A(I,I)))
  IP1 = K+2
  DO 180 I=IP1,N
  IM1 = I-1
  DO 180 J=K1,IM1
180 A(I,J)=A(I,J)-A(I,K)*DCONJG(A(J,K))-A(I,I)*DCONJG(A(J,K))
190 CONTINUE
  W(NM1,2)=CDABS(A(N,NM1))
  A(NM1,NM1)=A(N,NM1)/W(NM1,2)
C COMPUTE EIGENVALUES BY BISECTION METHOD
  CALL ERRSET(207,256,-1,1)
  R=DMAX1((DABS(W(1,1))+DABS(W(1,2))), (DABS(W(NM1,2))+DABS(W(N,1))))
  DO 200 I=2,NM1
  T=DABS(W(I-1,2))+DABS(W(I,1))+DABS(W(I,2))
  IF(T.GT.R) R=T
200 CONTINUE
  EPS1=R*0.1D-15
  EPS2=R*EPS
  DO 210 I=1,NM1
210 W(I,3)=W(I,2)**2
  IF(NE.LT.O) R=-R
  F=R
  DO 220 I=1,NEA
220 E(I)=-R
  DO 300 K=1,NEA
  D=E(K)
230 T = 0.5D0*(D+F)
  IF(DABS(D-T).LE.EPS2 .OR. DABS(F-T).LE.EPS2 ) GO TO 300
  J=0
  I=1
240 Q=W(I,1)-T
250 IF(Q.GE.O.) J=J+1
  IF(Q.EQ.O.) GO TO 260
  I=I+1
  IF(I.GT.N) GO TO 270
  CALL OVERFL(L)
  Q=W(I,1)-T-W(I-1,3)/Q
  CALL OVERFL(L)
```

```
      IF(L.NE.1) GO TO 250
      J=J+1
      I=I-1
260  I=I+2
      IF(I.LE.N) GO TO 240
270  IF(NE.LT.0) J=N-J
      IF(J.GE.K) GO TO 280
      F=T
      GO TO 230
280  D=T
      M=MINO(J,NEA)
      DO 290 I=K,M
290  E(I)=T
      GO TO 230
300  E(K)=T
C  COMPUTE EIGENVECTORS BY INVERSE ITERATION
310  CALL ERRSET(207, 10, 5,2)
      IF(NV.EQ.0) GO TO 900
      MM=584287
      CALL ERRSET(207,256,-1,1)
      DO 490 I=1,NVA
      DO 320 J=1,N
      W(J,3)=W(J,1)-E(I)
      W(J,4)=W(J,2)
320  W(J,7) = 1.0D0
      SW=.FALSE.
C  REDUCE TO TRIANGULAR FORM
      DO 340 J=1,NM1
      IF(DABS(W(J,3)).LT.DABS(W(J,2))) GO TO 330
      IF(W(J,3).EQ.0.) W(J,3)=1.0D-30
      W(J,6)=W(J,2)/W(J,3)
      LW(J)=.FALSE.
      W(J+1,3)=W(J+1,3)-W(J,6)*W(J,4)
      W(J,5)=0.
      GO TO 340
330  W(J,6)=W(J,3)/W(J,2)
      LW(J)=.TRUE.
      W(J,3)=W(J,2)
      T=W(J,4)
      W(J,4)=W(J+1,3)
      W(J,5)=W(J+1,4)
      W(J+1,3)=T-W(J,6)*W(J,4)
      W(J+1,4)=-W(J,6)*W(J,5)
340  CONTINUE
      IF(W(N,3).EQ.0.) W(N,3)=1.0D-30
C  BEGIN BACK SUBSTITUTION
      IF(I.EQ.1 .OR. DABS(E(I)-E(I-1)).GE.EPS1) GO TO 360
C  GENERATE RANDOM NUMBERS
      DO 350 J=1,N
      MM=MM*48828125
350  W(J,7)=FLOAT(MM)*0.4656613E-9
360  CALL OVERFL(L)
      T=W(N,7)
      R=W(N-1,7)
370  W(N,7)=T/W(N,3)
      W(N-1,7)=(R-W(N-1,4)*W(N,7))/W(N-1,3)
      CALL OVERFL(L)
```

```
      IF(L.NE.1) GO TO 390
      DO 380 J=1,N2
380  W(J,7)=W(J,7)*1.0D-5
      T=T*1.0D-5
      R=R*1.0D-5
      GO TO 370
390  IF(N.EQ.2) GO TO 440
      K=N2
400  T=W(K,7)
410  W(K,7)=(T-W(K,4)*W(K+1,7)-W(K,5)*W(K+2,7))/W(K,3)
      CALL OVERFL(L)
      IF(L.NE.1) GO TO 430
      DO 420 J=1,N
420  W(J,7)=W(J,7)*1.0D-5
      T=T*1.0D-5
      GO TO 410
430  K=K-1
      IF(K) 440,440,400
440  IF(SW) GO TO 470
      SW=.TRUE.
      DO 460 J=1,NM1
      IF(LW(J)) GO TO 450
      W(J+1,7)=W(J+1,7)-W(J,6)*W(J,7)
      GO TO 460
450  T=W(J,7)
      W(J,7)=W(J+1,7)
      W(J+1,7)=T-W(J,6)*W(J+1,7)
460  CONTINUE
      GO TO 360
470  DO 480 J=1,N
480  V(J,I)=W(J,7)
490  CONTINUE
C   BEGIN BACK TRANSFORMATION (1)
      CR = 1.0D0
      DO 500 J=2,N
      CR=CR*A(J-1,J-1)
      DO 500 I=1,NVA
500  V(J,I)=V(J,I)*CR
C   BEGIN BACK TRANSFORMATION (2)
      CALL ERRSET(207, 10, 5,2)
      IF(N.EQ.2) GO TO 600
      DO 590 I=1,NVA
      K=N2
550  CR=-A(K+1,K)*DCONJG(A(K,K))*W(K,2)
      IF(DREAL(CR).EQ.0.0 .AND. DIMAG(CR).EQ.0.0) GO TO 580
      CR = 1.0D0/CR
      CS=(0.,0.)
      IP1 = K+1
      DO 560 J=IP1,N
560  CS=CS+DCONJG(A(J,K))*V(J,I)
      CR=CR*CS
      DO 570 J=IP1,N
570  V(J,I)=V(J,I)-CR*A(J,K)
580  K=K-1
      IF(K.GE.1) GO TO 550
590  CONTINUE
600  CONTINUE
```

```

C NORMALIZE EIGENVECTORS
C NORMALIZE AS MAXIMUM ELEMENT=1
  DO 620 I=1,NVA
    T=DABS(DREAL(V(1,I)))+DABS(DIMAG(V(1,I)))
    K=1
    DO 610 J=2,N
      R=DABS(DREAL(V(J,I)))+DABS(DIMAG(V(J,I)))
      IF(T.GE.R) GO TO 610
      T=R
      K=J
610 CONTINUE
    CR = 1.0D0/V(K,I)
    DO 620 J=1,N
620 V(J,I)=V(J,I)*CR
    IF(NV.LT.0) GO TO 900
C ORTHOGONALIZE AS NORM=1
  DO 680 I=1,NVA
    IF(I.EQ.1 .OR. DABS(E(I)-E(I-1)).GT.EPS) GO TO 650
    IM1 = I-1
    DO 640 J=M,IM1
      CS=(0.,0.)
    DO 630 K=1,N
630 CS=CS+DCONJG(V(K,J))*V(K,I)
    DO 640 K=1,N
640 V(K,I)=V(K,I)-CS*V(K,J)
    GO TO 660
650 M=I
C NORMALIZE AS NORM=1
660 S=0.
    DO 670 J=1,N
670 S=S+DREAL(V(J,I))**2+DIMAG(V(J,I))**2
    T = DSQRT(1.0D0/S)
    DO 680 J=1,N
680 V(J,I)=V(J,I)*T
900 RETURN
C PRINT ERROR MESSAGE
910 WRITE(6,1000) N,NE
    GO TO 900
920 WRITE(6,1100) NV,NE,N,N1
    GO TO 900
1000 FORMAT(1H,'(SUBR. DEIGCH) N=',I5,',NE=',I5,' N SHOULD BE GREATER
1 THAN ZERO AND NE SHOULD BE NON-ZERO. RETURN WITH NO CALCULATION.
2 ')
1100 FORMAT(1H,'(SUBR. DEIGCH) NV=',I5,',NE=',I5,',N=',I5,',N1=',I5,
1 ' NV,NE,N,N1 SHOULD SATISFY THE FOLLOWING INEQUALITIES, !NV! <= !N
2E! <= N <= N1 .' /1H ,'RETURN WITH NO CALCULATION.')
    END
    SUBROUTINE ERRSET(I,J,K,L)
    RETURN
    END
    SUBROUTINE OVERFL(L)
    RETURN
    END

```


B.5 状態密度計算プログラム

```
c
c      状態密度計算プログラム計算プログラム
c
c      programed by T.Takeya
c
c      date 9.5.10
c
c      implicit real*8(a-h,p-z)
c      parameter(nnn=180,nj=200,n=40,njj=(nj+1))
c      dimension ejou(nnn),ekj(nnn),eee(3*n,njj)
c
c      do i=1,nnn
c      ejou(i)=0.0d0
c      end do
c      dw=10.0d0
c
c      do i=1,nnn
c      ekj(i)=float(i-1)*dw
c      end do
c
c      open(61,file='p10')
c      do j=1,njj
c      do i=1,3*n
c      read(61,*)rt,eee(i,j)
c      end do
c      end do
c      close(61)
c
c
c
c      do j=1,nnn
c      k=0
c      do i=1,3*n
c      do jj=1,njj
c      if((eee(i,jj).ge.float((j-1)*dw).and.
c      &(eee(i,jj).lt.float(j)*dw)) then
c      k=k+1
c      ejou(j)=dfloat(k)
c      endif
c      end do
c      end do
c      if(j.lt.3) then
c      write(*,*)ejou(j),ekj(j)
c      endif
c      end do
c
c
c
c      file の出力(='jotai1')
c
c      open(60,file='jotai1')
c      do i=1,nnn
c      write(60,4)ekj(i),ejou(i)*1.0d2/float(nj+1)/dfloat(n)/dw
c      end do
c      4 format(E10.5,f10.5)
c      close(60)
```

```
c
c  積分値の確かめ
c
  ppp=0.0d0
  do i=1,nnn
    if((ppp.eq.1).or.(ppp.eq.nnn)) then
      ppp=ppp+ejou(i)/float(nj+1)/dfloat(n)/dw
    else
      ppp=ppp+2.0d0*(ejou(i)/float(nj+1)/dfloat(n)/dw)
    endif
  end do
  write(*,*) ppp
  write(*,*) ' 積分値 =', dw/2.0d0*ppp
stop
end
```

B.6 立体角ラマン強度計算プログラム

1. 場所: file = takeya/for/tube/raman/r-it.f:
2. 実行方法: 螺旋度 (n,m) の時
ディレクトリ takeya/for/tube/raman/ に移動し、display 上で
nr n-m を入力する。
3. 入力ファイル: n-m.xyz (xyz 座標 (xmol)) = tube-xyz.f
:n-m.near (最近接データ) = saikn.f
:n-m.dat2 (固有値、固有ベクトル) = n-m.dat2 = tu-phonon1.f
4. 出力ファイル: n-m-i.dat (ラマン強度のデータ)

B.7 立体角ラマン強度計算プログラム

```

c
c   file=~takeya/for/tube/raman/r-it.f
c
c   ラマン強度計算プログラム (立体角ラマン強度)
c
c   入力ファイル: jobname.xyz   xyz 座標 (xmol)
c                   :jobname-p.dat P ベクトル
c   出力ファイル: jobname-i.dat ラマン強度
c
c   na: 原子数   x,y,z: 原子の x,y,z 座標   e: 元素記号
c   r0: n 番目の原子から最近接原子への vector
c   cai: f 番目の mode における n 番目の原子の固有 vector
c       cai(i,j,k)-> i: x=1,y=2,z=3   j: 原子の番号 n   k: 振動モード番号 f
c   p: Raman 強度を求めるための行列
c       p(i,j,k)-> i,j: x=1,y=2,z=3   k: 振動モード番号 f
c   d: 関数
c   alpha1 ~ 3 論文の値
c       alpha1:  '||-  '
c                   -
c       alpha2:  2  '+  '||
c                   -
c       alpha3:  ||-  |
c                   -
c
c ----- 初期設定 -----
c
c   implicit real*8(a-h,o-z)
c   character*16 jobname
c   character*8 f
c   n 個まで O.K.
c   character*8 e
c   parameter (n=600,mmk=30,nnk=30,ink=mmk*nnk,nl=40,ml=450)
c   dimension p(3,3,3*n),ei(3*n),th(mmk),phi(nnk)
c   dimension ai(3*n),ss(3,mmk,nnk),tt(3,mmk,nnk),f2(mmk)
c   dimension aii2(3*n,mmk,nnk),aik2(3*n,mmk),
c   &aii1(3*n,mmk,nnk),aik1(3*n,mmk),aikp(3*n,mmk)
c   dimension p(3,3,3*n),x(n),y(n),z(n),e(n)
c   dimension r0(3,n,3),cai(3,n,3*n),d(3,3),r0h(3,n,3),r2(n,3)
c   dimension is(n,4),isi(n,3),caih(3,n,3*n),cai2(n,3*n)
c   dimension a3(ml)
c   call getarg(1,jobname)
c   job=index(jobname,' ')-1
c   write(*,*) ' プログラム スタート !!'

```

```
      a1=4.0d0
      a2=4.5d0
      a3=0.04d0
c
c      VV=1,VH=2
c
      kv=1
c
c      a3=0.0d0
c      関数の初期設定

      do i=1,3
        do j=1,3
          d(i,j)=0.0
        end do
      end do

      do i=1,3
        d(i,i)=1.0
      end do

c -----
c ----- Data 入力 -----
c ----- xyz -----
      open(61,file=jobname(:job)//'.xyz')
      read(61,*) na
      if(n.lt.na) then
        write(*,*) 'n is smaller than na !!'
        stop
      end if
      do i=1,na
        read(61,*) e(i),x(i),y(i),z(i)
      end do
      close(61)
c -----
c ----- 最近接 -----
      open(62,file=jobname(:job)//'.near')
      read(62,*) t
c      write(*,*) t
      do i=1,na
        read(62,*) (is(i,j),j=1,4),(isi(i,j),j=1,3)
      end do
      close(62)
c -----
c ----- 固有ベクトル -----
      o=(3*na)/8
      n1=int(o)
      n2=mod(3*na,8)

c      write(*,*) n1,n2

      open(63,file=jobname(:job)//'.dat2')
      do i=1,n1
        read(63,*) (ei((i-1)*8+1),l=1,8)
        do j=1,na
          do k=1,3
```

```

        read(63,*) (cai(k,j,(i-1)*8+1),l=1,8)
    end do
end do
end do

if(n2.eq.0) then
    goto 200
endif

read(63,*) (ei((i-1)*8+1),l=1,n2)
do j=1,na
    do k=1,3
        read(63,*) (cai(k,j,n1*8+i),i=1,n2)
    end do
end do
close(63)

c -----

200 write(*,*) ' データ読み込み終了 !!'
c -----

c --- R0 vector を求める ---
c isi=-1 なら z=z-t
c isi=1 なら z=z+t

do i=1,na
    do j=1,3
        if(isi(i,j).eq.-1) then
            r0(1,i,j)=x(is(i,j+1))-x(i)
            r0(2,i,j)=y(is(i,j+1))-y(i)
            r0(3,i,j)=(z(is(i,j+1))-t)-z(i)
c            write(*,*) r0(1,i,j),r0(2,i,j),r0(3,i,j)
        endif
        if(isi(i,j).eq.1) then
            r0(1,i,j)=x(is(i,j+1))-x(i)
            r0(2,i,j)=y(is(i,j+1))-y(i)
            r0(3,i,j)=(z(is(i,j+1))+t)-z(i)
c            write(*,*) r0(1,i,j),r0(2,i,j),r0(3,i,j)
        endif
        if(isi(i,j).eq.0) then
            r0(1,i,j)=x(is(i,j+1))-x(i)
            r0(2,i,j)=y(is(i,j+1))-y(i)
            r0(3,i,j)=z(is(i,j+1))-z(i)
c            write(*,*) r0(1,i,j),r0(2,i,j),r0(3,i,j)
        endif
    end do
end do

c -----

c ---- R0,cai の unit vector を求める

do i=1,na
    do i1=1,3
        r=0.0
        do m=1,3
            r=r+r0(m,i,i1)*r0(m,i,i1)

```

```

        end do
        r2(i,i1)=sqrt(r)
        write(70,*) r2(i,i1)
    end do
end do

do i=1,na
    do i1=1,3
        r=0.0
        do m=1,3
            r0h(m,i,i1)=r0(m,i,i1)/r2(i,i1)
        end do
c       r=r0h(1,i,i1)**2+r0h(2,i,i1)**2+r0h(3,i,i1)**2
c       r1=sqrt(r)
c       write(*,*) r1
    end do
end do

do k=1,3*na
    do l=1,na
        c=0.0
        do i=1,3
            c=c+cai(i,l,k)*cai(i,l,k)
        end do
        cai2(l,k)=sqrt(c)
    end do
end do

do k=1,3*na
    do l=1,na
        do i=1,3
            caih(i,l,k)=cai(i,l,k)/cai2(l,k)
        end do
    end do
end do
c -----
c ----- P 初期化 -----

do i=1,3
    do j=1,3
        do k=1,3*na
            p(i,j,k)=0.0
        end do
    end do
end do
c -----
c ----- P の計算 -----

do 10 k=1,3*na
    do 20 j=1,3
        do 30 i=1,3
            do 40 l=1,na
                do 50 m=1,3

```

```

c -- 内積 and 行列の大きさの計算 --

```

```

sumrk1=0.0
sumrk2=0.0
do i1=1,3
  sumrk1=sumrk1+r0h(i1,l,m)*cai(i1,l,k)
  sumrk2=sumrk2+r0h(i1,l,m)*cai(i1,l,k)
end do

c -----

c ---- 行列成分計算 ----
c      do ik=1,nnk
p1=(a2*sumrk2*d(i,j))/3.0
p2=a1*(r0h(i,l,m)*r0h(j,l,m)-(d(i,j)/3.0)*sumrk2
p3=r0h(i,l,m)*caih(j,l,k)+r0h(j,l,m)*caih(i,l,k)
p4=2.0*r0h(i,l,m)*r0h(j,l,m)*sumrk2
p(i,j,k)=p(i,j,k)-(p1+p2+(a3*(p3-p4))/r2(l,m))
c      end do
50          continue
40          continue
30          continue
20          continue
10          continue
c
c
c Pベクトルの出力
c
c
c      open(72,file='p-vec')
c      do in=1,3*na
c          write(72,108) ei(in)
c          do ix=1,3
c              write(72,109) (p(ix,iy,in),iy=1,3)
c          end do
c      end do
108 format(f10.2)
109 format(3f10.5)
c      close(72)
c
c
c      romax=400
c      romin=10
c      open(72,file='p-vec1')
c      do in=1,3*na
c          if((ei(in).le.romax).and.(ei(in).ge.romin)) then
c              write(72,118) ei(in)
c              do ix=1,3
c                  write(72,119) (p(ix,iy,in),iy=1,3)
c              end do
118 format(f10.2)
119 format(3f10.5)
c          endif
c      end do
c      close(72)
c ---- P の計算 (ここまで) ----
c
c      強度計算
c

```

```
c
c
  hb=1.05459
  bk=1.38062
  t=300.0
  c=2.997925
  wi=19436.346
  Pi=4.0*atan(1.0)
c
c
c
c
c  初期化
c
  do i=1,mmk
    do jj=1,nnk
      do j=1,3
        ss(j,i,jj)=0.0
      end do
    end do
  end do
c
  do i=1,mmk
    do jj=1,nnk
      do j=1,3
        tt(j,i,jj)=0.0
      end do
    end do
  end do
c
c
  dphi=0.25/pi
  do L=1,mmk
    th(L)=abs(2.0*float(L)-1)/(2.0*float(mmk)) * Pi
    f2(L)=dphi*(cos(float(L-1)*Pi/float(mmk))
&-cos(float(L)*Pi/float(mmk)))
    end do
c
  do K=1,nnk
    phi(K)=(2.0*float(K))/float(nnk) * Pi
  end do
c
  do L=1,mmk
    do K=1,nnk
c
      ss(1,L,K)=-sin(th(L))
      ss(2,L,K)=0.0d0
      ss(3,L,K)=cos(th(L))
c
      tt(1,L,K)=cos(th(L))*cos(phi(K))
      tt(2,L,K)=-sin(phi(K))
      tt(3,L,K)=sin(th(L))*cos(phi(K))
    end do
  end do
c
c
c
```



```
c
c
c ----- Data 入力 -----
c ----- 原子総数 -----
      open(61,file=jobname(:job)//'.xyz')
      read(61,*) na
      if(n.lt.na) then
        write(*,*) 'n is smaller than na !!'
        stop
      end if
      close(61)
c -----
c ----- 強度計算 -----
      do i=1,3*na
        ai(i)=0.0
      end do
c
      do i=1,3*na
        do L=1,mmk
          aik2(i,L)=0.0
        end do
      end do
c
      do i=1,3*na
        do L=1,mmk
          aik1(i,L)=0.0
        end do
      end do
c
      do i=1,3*na
        do L=1,mmk
          do j=1,nnk
            aii1(i,L,j)=0.0
          end do
        end do
      end do
c
      do i=1,3*na
        do L=1,mmk
          do j=1,nnk
            aii2(i,L,j)=0.0
          end do
        end do
      end do
c
      do i=1,3*na
        if(ei(i).lt.0.1)then
          goto 100
        endif
c
          a1=1/(exp(hb*ei(i)*c/(bk*t*10))-1)
c
c   VH 方向
c
      do 194 L=1,mmk
        do LK=1,nnk
          agg=0.0
```

```

        do j=1,3
            do kk=1,3
                aii2(i,L,LK)=ss(j,L,LK)*tt(kk,L,LK)*p(j,kk,i)+agg
                agg=aii2(i,L,LK)
            end do
        end do

        cc=aii2(i,L,LK)*aii2(i,L,LK)
        cd=(wi-ei(i))**3
        aii2(i,L,LK)=cd*(a1+1.0)*cc/ei(i)
    end do

c
    do jj=1,nnk
        aik2(i,L)=aii2(i,L,jj)+aik2(i,L)
    end do

194 continue

c
c VV 方向
    do 195 L=1,mmk
        do LK=1,nnk
            agg=0.0
            do j=1,3
                do kk=1,3
                    aii1(i,L,LK)=ss(j,L,LK)*ss(kk,L,LK)*p(j,kk,i)
                    &/dfloat(nnk)+agg
                    agg=aii1(i,L,LK)
                end do
            end do

            cc=aii1(i,L,LK)*aii1(i,L,LK)
            cd=(wi-ei(i))**3
            aii1(i,L,LK)=cd*(a1+1.0)*cc/ei(i)
        end do

    c
        do jj=1,nnk
            aik1(i,L)=aii1(i,L,jj)+aik1(i,L)
        end do

195 continue

c
c
    do L=1,mmk
        if(kv.eq.1) then
            aikp(i,L)=aik1(i,L)
        endif
        if(kv.eq.2) then
            aikp(i,L)=aik2(i,L)
        endif

    c
        ai(i)=aikp(i,L)*sin(th(L))*pi/mmk*2.0*pi/nnk+ai(i)
    c
        &+aik2(i,L)*sin(th(L))*pi/mmk*2.0*pi/nnk
    end do

c
100 end do

c -----
    do i=1,3*na
        if(ei(i).ne.ei(i+1)) then
            goto 201
        else
            ai(i+1)=ai(i+1)+ai(i)
        end if
    end do

```

```

        endif
201  end do

c ----- Normalization -----
    aimax=0.0
    do i=1,3*na

        if(aimax.lt.ai(i)) then
            aimax=ai(i)
            ee=ei(i)
c            abc3=a3(ik)
        endif
    end do
    if(ee.le.200) then
        goto 129
    endif
    write(*,*)aimax,ee
122  continue
c
129  continue
    do i=1,3*na
        ai(i)=ai(i)/aimax
    end do

    write(*,*) '規格化なし=',aimax*1.0d-6
c    write(*,*) 'a3=',abc3
    write(*,*) 6350615.5620460/12771.985444200
c -----

c ----- Data 出力 -----
p
    open(72,file=jobname(:job)//'-i.dat')
    do i=1,3*na
        if(ei(i).ne.ei(i+1)) then
            write(72,1001) ei(i),ai(i)
        end if
    end do
    close(72)

c -----
1001 format(f10.4,f10.5)
    stop
end

```

B.8 端のある立体角ラマン強度計算プログラム

1. 場所:file= takeya/for/tube/raman/nr-it.f:
2. 実行方法: 例えば螺旋度 10T(10,10) の時
ディレクトリ takeya/for/tube/raman/ に移動し、display 上で
nr 10n10-10 を入力する。
3. 入力ファイル:10n10-10.xyz(xyz 座標 (xmol))=nanbai.f
:10n10-10.near(最近接データ)=nanbai.f
:10n10-10.dat2(固有値、固有ベクトル) =Ntu-phonon1.f
4. 出力ファイル:n-m-i.dat (ラマン強度のデータ)

```
c    raman-i ラマン強度計算プログラム
```

```

c      '96/8/13 作成
c      入力ファイル:jobname.xyz   xyz 座標 (xmol)
c              :jobname-p.dat P ベクトル
c      出力ファイル:jobname-i.dat ラマン強度
c
c      na: 原子数  x,y,z: 原子の x,y,z 座標  e: 元素記号
c      r0: n 番目の原子から最近接原子への vector
c      cai: f 番目の mode における n 番目の原子の固有 vector
c          cai(i,j,k)-> i: x=1,y=2,z=3  j: 原子の番号 n  k: 振動モード番号 f
c      p: Raman 強度を求めるための行列
c          p(i,j,k)-> i,j: x=1,y=2,z=3  k: 振動モード番号 f
c      d: 関数
c      alpha1 ~ 3 論文の値
c          alpha1:  '|-  '|
c                  -
c          alpha2: 2  '|+  '|
c                  -
c          alpha3:  '|-  |
c                  -
c
c ----- 初期設定 -----
c
c      implicit real*8(a-h,o-z)
c      character*16 jobname
c      character*8 f
c      n 個まで O.K.
c      character*8 e
c      parameter (n=1200,mmk=30,nnk=30,ink=mmk*nnk,nl=40,ml=450)
c
c      dimension p(3,3,3*n),ei(3*n),th(mmk),phi(nnk)
c      dimension ai(3*n),ss(3,mmk,nnk),tt(3,mmk,nnk),f2(mmk)
c      dimension aii2(3*n,mmk,nnk),aik2(3*n,mmk),
c      &aii1(3*n,mmk,nnk),aik1(3*n,mmk),aikp(3*n,mmk)
c      dimension p(3,3,3*n),x(n),y(n),z(n),e(n)
c      dimension r0(3,n,3),cai(3,n,3*n),d(3,3),r0h(3,n,3),r2(n,3)
c      dimension is(n,4),isi(n,3),caih(3,n,3*n),cai2(n,3*n),aca(3,n)
c      dimension a3(ml).aca(3,n)
c      call getarg(1,jobname)
c      job=index(jobname,' ')-1
c      write(*,*) ' プログラム スタート!!'
c      a1=4.5d0
c      a2=4.0d0
c      a3=0.04d0
c
c      VV=1,VH=2
c
c      kv=1
c
c      a3=0.0d0
c      関数の初期設定
c
c      do i=1,3
c          do j=1,3
c              d(i,j)=0.0
c          end do
c      end do

```

```
do i=1,3
  d(i,i)=1.0
end do

c -----

c ---- Data 入力 -----
c ----- xyz -----
open(61,file=jobname(:job)//'.xyz')
read(61,*) na
if(n.lt.na) then
  write(*,*) 'n is smaller than na !!'
  stop
end if
do i=1,na
  read(61,*) e(i),x(i),y(i),z(i)
end do
close(61)

c -----
c ----- 最近接 -----
open(62,file=jobname(:job)//'.near')
c   read(62,*) t
c   write(*,*) t
do i=1,na
  read(62,*) (is(i,j),j=1,4),(isi(i,j),j=1,3)
end do
close(62)

c -----
c ----- 固有ベクトル -----
o=(3*na)/8
n1=int(o)
n2=mod(3*na,8)

c   write(*,*) n1,n2

open(63,file=jobname(:job)//'.dat2')
do i=1,n1
  read(63,*) (ei((i-1)*8+1),l=1,8)
  do j=1,na
    do k=1,3
      read(63,*) (cai(k,j,(i-1)*8+1),l=1,8)
    end do
  end do
end do

if(n2.eq.0) then
  goto 200
endif

read(63,*) (ei((i-1)*8+1),l=1,n2)
do j=1,na
  do k=1,3
    read(63,*) (cai(k,j,n1*8+i),i=1,n2)
  end do
end do
close(63)

c -----
```

```
200 write(*,*) ' データ読み込み終了 !!'
c -----
c
      do j=1,na
        do k=1,3
          aca(k,j)=0.0d0
        end do
      end do

c
c -----
c   補足
c
      bai=40.0
      do i=1,na*3
        if((i.eq.2318).or.(i.eq.2321).or.
&(i.eq.2327).or.
&(i.eq.2333)) then
          do j=1,na
            do k=1,3
              aca(k,j)=cai(k,j,i)*bai+aca(k,j)
            end do
          end do
        endif
      end do

c
      open(60,file='v.xyz')
      write(60,*) na
      write(60,*) ' '
      do j=1,na
        write(60,661) x(j),y(j),z(j),(aca(k,j),k=1,3)
661 format('C',6f10.5)
      end do
      close(60)

c
c   何番目の固有値か
c   write(*,*)'ii='
c   read(*,*)ii

c
c
      open(60,file='20n-z.dat')
      do j=1,na,40
        czz=0.0
        czz2=0.0

c
      do i=1,na
        if(z(j).eq.z(i)) then
          czz=cai(3,i,ii)+czz
        endif
      end do
      write(60,662) z(j),czz

c
      do i=1,na
        if(z(j+1).eq.z(i)) then
          czz2=cai(3,i,ii)+czz2
        endif
      end do
```

```

        write(60,662) z(j+1),czz2
c
662      format(2f10.5)
        end do
        close(60)
c
c
        open(60,file='20n-d.dat')
        do j=40,na,40
            if(cai(1,j,ii).ge.1.0d-16) then
czz=sqrt(cai(2,j,ii)**2+cai(1,j,ii)**2)*20.0
            else
czz=-sqrt(cai(2,j,ii)**2+cai(1,j,ii)**2)*20.0
            endif
        write(60,663) z(j),czz
        write(*,*)x(j)
c
c
        if(cai(1,j-1,ii).ge.1.0d-16) then
czz=sqrt(cai(2,j-1,ii)**2+cai(1,j-1,ii)**2)*20.0
            else
czz=-sqrt(cai(2,j-1,ii)**2+cai(1,j-1,ii)**2)*20.0
            endif
        write(60,663) z(j-1),czz
        write(*,*)x(j-1)
663      format(2f10.5)
        end do
        close(60)
write(*,*) 'd-end dayo'
c   補足
c
c
c   open(60,file='gg-eval')
c   do i=1,na*3
c   write(60,*)i,ei(i)
c   end do
c   close(60)
c-----
c
c
c --- R0 vector を求める ---
c isi=-1 なら z=z-t
c isi=1  なら z=z+t

        do i=1,na
            do j=1,3
                if(isi(i,j).eq.0) then
                    r0(1,i,j)=0.0d0
                    r0(2,i,j)=0.0d0
                    r0(3,i,j)=0.0d0
c
                    write(*,*) r0(1,i,j),r0(2,i,j),r0(3,i,j)
                endif
                if(isi(i,j).eq.1) then
                    r0(1,i,j)=x(is(i,j+1))-x(i)
                    r0(2,i,j)=y(is(i,j+1))-y(i)
                    r0(3,i,j)=z(is(i,j+1))-z(i)
c
                    write(*,*) r0(1,i,j),r0(2,i,j),r0(3,i,j)

```

```
        endif
      end do
    end do
c -----

c ---- R0,cai の unit vector を求める

    do i=1,na
      do i1=1,3
        r=0.0
        do m=1,3
          r=r+r0(m,i,i1)*r0(m,i,i1)
        end do
        r2(i,i1)=sqrt(r)
        write(70,*) r2(i,i1)
      end do
    end do

    do i=1,na
      do i1=1,3
        if(isi(i,i1).eq.1) then
          r=0.0
          do m=1,3
            r0h(m,i,i1)=r0(m,i,i1)/r2(i,i1)
          end do
        endif
c      r=r0h(1,i,i1)**2+r0h(2,i,i1)**2+r0h(3,i,i1)**2
c      r1=sqrt(r)
c      write(*,*) r1
    end do
  end do

  do k=1,3*na
    do l=1,na
      c=0.0
      do i=1,3
        c=c+cai(i,l,k)*cai(i,l,k)
      end do
      cai2(l,k)=sqrt(c)
    end do
  end do

  do k=1,3*na
    do l=1,na
      do i=1,3
        caih(i,l,k)=cai(i,l,k)/cai2(l,k)
      end do
    end do
  end do
c -----

c ---- P 初期化 -----

  do i=1,3
    do j=1,3
      do k=1,3*na
        p(i,j,k)=0.0
      end do
    end do
  end do
```



```

        end do
    end do
end do
c -----
c ----- P の計算 -----

do 10 k=1,3*na
    do 20 j=1,3
        do 30 i=1,3
            do 40 l=1,na
                do 50 m=1,3

c -- 内積 and 行列の大きさの計算 --

                sumrk1=0.0
                sumrk2=0.0
                do i1=1,3
                    sumrk1=sumrk1+r0h(i1,l,m)*cai(i1,l,k)
                    sumrk2=sumrk2+r0h(i1,l,m)*cai(i1,l,k)
                end do

c -----

c ---- 行列成分計算 ----
c      do ik=1,nnk
c      if(isi(l,m).eq.1) then

                p1=(a2*sumrk2*d(i,j))/3.0
                p2=a1*(r0h(i,l,m)*r0h(j,l,m)-(d(i,j)/3.0))*sumrk2
                p3=r0h(i,l,m)*caih(j,l,k)+r0h(j,l,m)*caih(i,l,k)
                p4=2.0*r0h(i,l,m)*r0h(j,l,m)*sumrk2
                p(i,j,k)=p(i,j,k)-(p1+p2+(a3*(p3-p4))/r2(l,m))
c      end do
                endif

50          continue
40          continue
30          continue
20          continue
10          continue
c
c
c P ベクトルの出力
c
c
c      open(72,file='p-vec')
c      do in=1,3*na
c          write(72,108) ei(in)
c          do ix=1,3
c              write(72,109) (p(ix,iy,in),iy=1,3)
c          end do
c      end do
108 format(f10.2)
109 format(3f10.5)
c      close(72)
c
c
c      romax=400

```

```
    romin=10
    open(72,file='p-vec1')
    do in=1,3*na
        if((ei(in).le.romax).and.(ei(in).ge.romin)) then
            write(72,118) ei(in)
            do ix=1,3
                write(72,119) (p(ix,iy,in),iy=1,3)
            end do
118 format(f10.2)
119 format(3f10.5)
            endif
        end do
        close(72)
c ----- P の計算 (ここまで) -----
c
c   強度計算
c
c
c
c
c   hb=1.05459
c   bk=1.38062
c   t=300.0
c   c=2.997925
c   wi=19436.346
c   Pi=4.0*atan(1.0)
c
c
c
c
c   初期化
c
c   do i=1,mmk
c       do jj=1,nnk
c           do j=1,3
c               ss(j,i,jj)=0.0
c           end do
c       end do
c   end do
c
c
c   do i=1,mmk
c       do jj=1,nnk
c           do j=1,3
c               tt(j,i,jj)=0.0
c           end do
c       end do
c   end do
c
c
c   dphi=0.25/pi
c       do L=1,mmk
c           th(L)=abs(2.0*float(L)-1)/(2.0*float(mmk)) * Pi
c           f2(L)=dphi*(cos(float(L-1)*Pi/float(mmk))
c &-cos(float(L)*Pi/float(mmk)))
c       end do
c
c
c       do K=1,nnk
c           phi(K)=(2.0*float(K))/float(nnk) * Pi
```

```

        end do
c
        do L=1,mmk
            do K=1,nnk
c
                ss(1,L,K)=-sin(th(L))
                ss(2,L,K)=0.0d0
                ss(3,L,K)=cos(th(L))
c
                tt(1,L,K)=cos(th(L))*cos(phi(K))
                tt(2,L,K)=-sin(phi(K))
                tt(3,L,K)=sin(th(L))*cos(phi(K))
            end do
        end do
c
c
c
c
c
c ----- Data 入力 -----
c ----- 原子総数 -----
        open(61,file=jobname(:job)//'.xyz')
        read(61,*) na
        if(n.lt.na) then
            write(*,*) 'n is smaller than na !!'
            stop
        end if
        close(61)
c -----
c ----- 強度計算 -----
        do i=1,3*na
            ai(i)=0.0
        end do
c
        do i=1,3*na
            do L=1,mmk
                aik2(i,L)=0.0
            end do
        end do
c
        do i=1,3*na
            do L=1,mmk
                aik1(i,L)=0.0
            end do
        end do
c
        do i=1,3*na
            do L=1,mmk
                do j=1,nnk
                    aii1(i,L,j)=0.0
                end do
            end do
        end do
c
        do i=1,3*na
            do L=1,mmk
                do j=1,nnk
```

```

        aii2(i,L,j)=0.0
    end do
end do
end do
c
do i=1,3*na
    if(ei(i).lt.0.1)then
        goto 100
    endif
c
        a1=1/(exp(hb*ei(i)*c/(bk*t*10))-1)
c
c    VH 方向
c
        do 194 L=1,mmk
            do LK=1,nnk
                agg=0.0
                do j=1,3
                    do kk=1,3
                        aii2(i,L,LK)=ss(j,L,LK)*tt(kk,L,LK)*p(j,kk,i)+agg
                        agg=aii2(i,L,LK)
                    end do
                end do
                cc=aii2(i,L,LK)*aii2(i,L,LK)
                cd=(wi-ei(i))**3
                aii2(i,L,LK)=cd*(a1+1.0)*cc/ei(i)
            end do
c
            do jj=1,nnk
                aik2(i,L)=aii2(i,L,jj)+aik2(i,L)
            end do
194    continue
c
c    VV 方向
        do 195 L=1,mmk
            do LK=1,nnk
                agg=0.0
                do j=1,3
                    do kk=1,3
                        aii1(i,L,LK)=ss(j,L,LK)*ss(kk,L,LK)*p(j,kk,i)
                        &/dfloat(nnk)+agg
                        agg=aii1(i,L,LK)
                    end do
                end do
                cc=aii1(i,L,LK)*aii1(i,L,LK)
                cd=(wi-ei(i))**3
                aii1(i,L,LK)=cd*(a1+1.0)*cc/ei(i)
            end do
c
            do jj=1,nnk
                aik1(i,L)=aii1(i,L,jj)+aik1(i,L)
            end do
195    continue
c
c
        do L=1,mmk
            if(kv.eq.1) then

```

```
        aikp(i,L)=aik1(i,L)
        endif
        if(kv.eq.2) then
            aikp(i,L)=aik2(i,L)
        endif
c
        ai(i)=aikp(i,L)*sin(th(L))*pi/mmk*2.0*pi/nnk+ai(i)
c      &+aik2(i,L)*sin(th(L))*pi/mmk*2.0*pi/nnk
        end do
c
100  end do
c -----
        do i=1,3*na
            if(ei(i).ne.ei(i+1)) then
                goto 201
            else
                ai(i+1)=ai(i+1)+ai(i)
            endif
201  end do

c ----- Normalization -----
        aimax=0.0
        do i=1,3*na

            if(aimax.lt.ai(i)) then
                aimax=ai(i)
                ee=ei(i)
c              abc3=a3(ik)
            endif
        end do
        if(ee.le.200) then
            goto 129
        endif
        write(*,*)aimax,ee
122  continue
c
129  continue
        do i=1,3*na
            ai(i)=ai(i)/aimax
        end do

        write(*,*) '規格化なし=',aimax*1.0d-6
c      write(*,*) 'a3=',abc3
        write(*,*) 6350615.5620460/12771.985444200
c -----

c ----- Data 出力 -----

        open(72,file=jobname(:job)//'-i.dat')
        do i=1,3*na
            if(ei(i).ne.ei(i+1)) then
                write(72,1001) ei(i),ai(i)
            end if
        end do
        close(72)
c -----
```

```
1001 format(f10.4,f10.2)
      stop
      end
```

B.9 ラマン強度計算プログラム (角度依存性)

場所:file= takeya/for/tube/raman/r-itsxy.f 2. 実行方法: 例えば螺旋度 (n,m) の時
ディレクトリ takeya/for/tube/raman/ に移動し、display 上で

```
"f77 -o rs r-itsxy.f"
```

```
"rs n-m"
```

をする。

を入力する。

3. 入力ファイル:n-m.xyz(xyz 座標 (xmol))=tube-xyz.f

:n-m.near(最近接データ)=saikn.f

:n-m.dat2(固有値、固有ベクトル)=tu-phonon1.f 4. 出力ファイル:n-m-is.dat(ラマン強度角度依存性のデータ)

```
c
c      file=~takeya/for/tube/raman/r-itsxy.f
c
c      ラマン強度計算プログラム (角度依存性)
c      '96/8/13 作成
c      入力ファイル:jobname.xyz   xyz 座標 (xmol)
c                  :jobname-p.dat P ベクトル
c      出力ファイル:jobname-i.dat ラマン強度
c
c      na: 原子数  x,y,z: 原子の x,y,z 座標  e: 元素記号
c      r0: n 番目の原子から最近接原子への vector
c      cai: f 番目の mode における n 番目の原子の固有 vector
c           cai(i,j,k)-> ai(116,L),i: x=1,y=2,z=3  j: 原子の番号 n  k: 振動モード番号 f
c      p: Raman 強度を求めるための行列
c           p(i,j,k)-> i,j: x=1,y=2,z=3  k: 振動モード番号 f
c      d: 関数
c      alpha1 ~ 3 論文の値
c      alpha1:  '|-  '|
c              -
c      alpha2: 2  '|+  '|
c              -
c      alpha3:  '|-  |
c              -
c
c ----- 初期設定 -----
c
c      implicit real*8(a-h,o-z)
c      character*16 jobname
c      character*8 f
c      n 個まで O.K.
c      character*8 e
c      parameter (n=300,mmk=500,nnk=2,ink=mmk*nnk,n1=40,m1=450)
c
c      dimension p(3,3,3*n),ei(3*n),th(mmk),phi(nnk)
c      dimension ai(3*n,mmk),ss(3,mmk),tt(3,mmk),f2(mmk)
c      dimension aihv(3*n,mmk),aivv(3*n,mmk)
c      dimension p(3,3,3*n),x(n),y(n),z(n),e(n)
c      dimension r0(3,n,3),cai(3,n,3*n),d(3,3),r0h(3,n,3),r2(n,3)
c      dimension is(n,4),isi(n,3),caih(3,n,3*n),cai2(n,3*n)
c      call getarg(1,jobname)
c      job=index(jobname,' ')-1
c      write(*,*) ' プログラム スタート!!'
c      a1=5.0d0
c      a2=4.5d0
c      a3=0.04d0
c
```

```
c   どの軸に回転するか ---y 軸;ksita=1   x 軸;ksita=2   z 軸;ksita=3
c
c   ksita=2
c
c   VV;kv=1   VH;kv=2
c
c   kv=1
c   関数の初期設定

c   do i=1,3
c     do j=1,3
c       d(i,j)=0.0d0
c     end do
c   end do

c   do i=1,3
c     d(i,i)=1.0d0
c   end do

c -----
c ----- Data 入力 -----
c ----- xyz -----
c   open(61,file=jobname(:job)//'.xyz')
c   read(61,*) na
c   if(n.lt.na) then
c     write(*,*) 'n is smaller than na !!'
c     stop
c   end if
c   do i=1,na
c     read(61,*) e(i),x(i),y(i),z(i)
c   end do
c   close(61)
c -----
c ----- 最近接 -----
c   open(62,file=jobname(:job)//'.near')
c   read(62,*) t
c   write(*,*) t
c   do i=1,na
c     read(62,*) (is(i,j),j=1,4),(isi(i,j),j=1,3)
c   end do
c   close(62)
c -----
c ----- 固有ベクトル -----
c   o=(3*na)/8
c   n1=int(o)
c   n2=mod(3*na,8)

c   write(*,*) n1,n2

c   open(63,file=jobname(:job)//'.dat2')
c   do i=1,n1
c     read(63,*) (ei((i-1)*8+1),l=1,8)
c     do j=1,na
c       do k=1,3
c         read(63,*) (cai(k,j,(i-1)*8+1),l=1,8)
c       end do
c     end do
c   end do
```



```

        end do
    end do

    if(n2.eq.0) then
        goto 200
    endif

    read(63,*) (ei((i-1)*8+1),l=1,n2)
    do j=1,na
        do k=1,3
            read(63,*) (cai(k,j,n1*8+i),i=1,n2)
        end do
    end do
    close(63)

c -----

    200 write(*,*) ' データ読み込み終了 !!'
c -----

c --- R0 vector を求める ---
c isi=-1 なら z=z-t
c isi=1 なら z=z+t

    do i=1,na
        do j=1,3
            if(isi(i,j).eq.-1) then
                r0(1,i,j)=x(is(i,j+1))-x(i)
                r0(2,i,j)=y(is(i,j+1))-y(i)
                r0(3,i,j)=(z(is(i,j+1))-t)-z(i)
c                write(*,*) r0(1,i,j),r0(2,i,j),r0(3,i,j)
            endif
            if(isi(i,j).eq.1) then
                r0(1,i,j)=x(is(i,j+1))-x(i)
                r0(2,i,j)=y(is(i,j+1))-y(i)
                r0(3,i,j)=(z(is(i,j+1))+t)-z(i)
c                write(*,*) r0(1,i,j),r0(2,i,j),r0(3,i,j)
            endif
            if(isi(i,j).eq.0) then
                r0(1,i,j)=x(is(i,j+1))-x(i)
                r0(2,i,j)=y(is(i,j+1))-y(i)
                r0(3,i,j)=z(is(i,j+1))-z(i)
c                write(*,*) r0(1,i,j),r0(2,i,j),r0(3,i,j)
            endif
        end do
    end do
c -----

c ---- R0,cai の unit vector を求める

    do i=1,na
        do i1=1,3
            r=0.0d0
            do m=1,3
                r=r+r0(m,i,i1)*r0(m,i,i1)
            end do
            r2(i,i1)=sqrt(r)

```

```

        write(70,*) r2(i,i1)
    end do
end do

do i=1,na
    do i1=1,3
        r=0.0d0
        do m=1,3
            r0h(m,i,i1)=r0(m,i,i1)/r2(i,i1)
        end do
c         r=r0h(1,i,i1)**2+r0h(2,i,i1)**2+r0h(3,i,i1)**2
c         r1=sqrt(r)
c         write(*,*) r1
    end do
end do

do k=1,3*na
    do l=1,na
        c=0.0d0
        do i=1,3
            c=c+cai(i,l,k)*cai(i,l,k)
        end do
        cai2(l,k)=sqrt(c)
    end do
end do

do k=1,3*na
    do l=1,na
        do i=1,3
            caih(i,l,k)=cai(i,l,k)/cai2(l,k)
        end do
    end do
end do
c -----
c ----- P 初期化 -----

do i=1,3
    do j=1,3
        do k=1,3*na
            p(i,j,k)=0.0d0
        end do
    end do
end do
c -----

c ----- P の計算 -----

do 10 k=1,3*na
    do 20 j=1,3
        do 30 i=1,3
            do 40 l=1,na
                do 50 m=1,3

```

c -- 内積 and 行列の大きさの計算 --

```

sumrk1=0.0

```

```

sumrk2=0.0
do i1=1,3
  sumrk1=ppsumrk1+r0h(i1,l,m)*cai(i1,l,k)
  sumrk2=sumrk2+r0h(i1,l,m)*cai(i1,l,k)
end do

c -----

c ---- 行列成分計算 ----
c   do ik=1,nnk
p1=(a2*sumrk2*d(i,j))/3.0d0
p2=a1*(r0h(i,l,m)*r0h(j,l,m)-(d(i,j)/3.0))*sumrk2
p3=r0h(i,l,m)*caih(j,l,k)+r0h(j,l,m)*caih(i,l,k)
p4=2.0*r0h(i,l,m)*r0h(j,l,m)*sumrk2
p(i,j,k)=p(i,j,k)-(p1+p2+(a3*(p3-p4))/r2(l,m))
c   end do
50           continue
40           continue
30           continue
20           continue
10           continue

c ---- P の計算 (ここまで) -----
c
c   強度計算
c
c
c
c   hb=1.05459d0
c   bk=1.38062d0
c   t=300.0d0
c   c=2.997925d0
c   wi=19436.346d0
c   Pi=4.0d0*atan(1.0d0)
c
c
c
c   初期化
c
c   do i=1,mmk
c     do j=1,3
c       ss(j,i)=0.0d0
c     end do
c   end do
c
c   do i=1,mmk
c     do j=1,3
c       tt(j,i)=0.0d0
c     end do
c   end do
c
c
c   dphi=0.25/pi
c   do L=1,mmk
c     th(L)=abs(1.0d0*dfloat(L)-1)/(2.0d0*dfloat(mmk)) * Pi
c     f2(L)=dphi*(cos(dfloat(L-1)*Pi/dfloat(mmk))

```



```
c
c -----
c ----- 強度計算 -----
  do i=1,3*na
    do L=1,mmk
      ai(i,L)=0.0d0
    end do
  end do

c
  do i=1,3*na
    do L=1,mmk
      aihv(i,L)=0.0d0
    end do
  end do

c
  do i=1,3*na
    do L=1,mmk
      aivv(i,L)=0.0d0
    end do
  end do

c
c
c
c
  do i=1,3*na
    if(ei(i).lt.0.1)then
      goto 100
    endif
  end do

c
      a1=1/(exp(hb*ei(i)*c/(bk*t*10))-1)

c
c  VH 方向
c
  do L=1,mmk
    agg=0.0d0
    do j=1,3
      do kk=1,3
        aihv(i,L)=ss(j,L)*tt(kk,L)*p(j,kk,i)+agg
        agg=aihv(i,L)
      end do
    end do

    cc=aihv(i,L)*aihv(i,L)
    cd=(wi-ei(i))**3
    aihv(i,L)=cd*(a1+1.0d0)*cc/ei(i)
  end do

c
c
c  VV 方向
  do L=1,mmk
    agg=0.0d0
    do j=1,3
      do kk=1,3
        aivv(i,L)=ss(j,L)*ss(kk,L)*p(j,kk,i)+agg
        agg=aivv(i,L)
      end do
    end do

    cc=aivv(i,L)*aivv(i,L)
```

```

        cd=(wi-ei(i))*3
        aivv(i,L)=cd*(a1+1.0)*cc/ei(i)
    end do

c
c
100 end do

c -----
    do L=1,mmk
c
c
        do i=1,3*na
c
c      HV o.r VV
c
            if(kv.eq.1) then
                ai(i,L)=aivv(i,L)
            endif
            if(kv.eq.2) then
                ai(i,L)=aihv(i,L)
            endif
c
c
c
            if(ei(i).ne.ei(i+1)) then
                goto 201
            else
                ai(i+1,L)=ai(i+1,L)+ai(i,L)
            endif
201 end do
    end do
c    if(ai(13,10).gt.ai(14,10))stop 'hen'
c ----- Data 出力 -----
    open(72,file=jobname(:job)//'-is.dat')
    moto=7
    dk=1.0d-12
c
    if(moto.ge.7) then
        do L=1,mmk
            write(72,1011) th(L)*180d0/pi,ai(8,L)*dk,ai(12,L)*dk
            &,ai(14,L)*dk,ai(95,L)*dk,ai(108,L)*dk,ai(110,L)*dk,
            &ai(116,L)*dk
        end do
    endif
c
    close(72)
c
c 各 角度のラマン強度 を求める。
c
    K=10
    sL=th(K)*180.0d0/pi
    write(*,*) sL
    open(72,file=jobname(:job)//'-iss.dat')
    do i=1,na*3
        if(ei(i).ne.ei(i+1)) then
            write(72,1001) ei(i),ai(i,K)

```

```
        end if
    end do
    close(72)
c
c -----
1001 format(f10.4,' ',f10.5)
1011 format(f10.5,7f25.5)
    stop
    end
```

著者の学外における発表実績

- 学会発表

- 第 12 回フラレン総合シンポジウム (平成九年 1 月)
カーボンナノチューブのフォノン分散関係
T. Takeya, R. Saito, T. Kimura G. Dresselhaus, M. S. Dresselhaus
- 日本物理学会 1997 年春の年会 (平成九年 3 月)
カーボンナノチューブのフォノン分散関係とラマン強度
T. Takeya, R. Saito, T. Kimura G. Dresselhaus, M. S. Dresselhaus
- 日本物理学会 1997 年秋の分科会 (平成九年 10 月)
カーボンナノチューブのラマン強度の半径依存性
T. Takeya, R. Saito, T. Kimura G. Dresselhaus, M. S. Dresselhaus
- 日本物理学会 1998 年春の年会 (平成 10 年 3 月)
カーボンナノチューブのラマン強度の端の効果
T. Takeya, R. Saito, T. Kimura G. Dresselhaus, M. S. Dresselhaus

- 論文発表

Raman Intensity of Single-Wall Carbon Nanotubes

R. Saito, T. Takeya, T. Kimura G. Dresselhaus, M. S. Dresselhaus,
Phys. Rev. B15(1998) 二月号