

第一原理擬ポテンシャル法による 電子状態計算プログラム

「Osaka2002_*nano*」

I 「基礎編」

第0.7版

大阪大学・産業科学研究所

白井光雲

18 Feb 2003

History
Rev. 0.7 04 March 2003

目次

序論	3
第1章 理論	6
1.1 密度汎関数法	6
1.2 局所密度近似	8
1.3 擬ポテンシャル近似	8
1.4 平面波展開	10
1.5 平面波のカットオフ	12
1.6 k 空間上の総和	13
1.7 Kohn-Sham 汎関数における共役勾配最小法	14
1.8 高速フーリエ変換	18
1.9 Hellmann-Feynman 力とストレス	19
第2章 ロードマップ	20
2.1 概略	20
2.2 計算環境と準備	21
2.2.1 機種ごとの対応	23
第3章 原子擬ポテンシャル	24
3.1 入力ファイル	24
3.2 実行	27
第4章 結晶の作成	29
4.1 結晶データ入力	29
4.2 <code>cryst</code> の実行	32
4.3 制御パラメータ	33
4.4 結晶のグラフィックス表示	34
第5章 電子の基底状態計算 (I)	35
5.1 <code>inip</code>	35
5.1.1 k サンプリング法	36
5.1.2 平面波カットオフ半径	39
5.2 <code>pwm</code> の動作	40
5.2.1 セルフコンシステント計算	40
5.2.2 解釈	44
5.2.3 再計算	49

5.2.4	電荷密度の表示	49
5.2.5	収束がうまくいかない場合	51
第 6 章	電子の基底状態計算 (II)	53
6.1	原子位置の最適化	53
6.1.1	計算の準備	53
6.1.2	結果の読み方	54
6.1.3	力の誤差に関して	55
6.1.4	収束に関する議論	56
6.2	格子定数の最適化	58
6.2.1	計算の準備	58
6.2.2	結果の読み方	59
6.2.3	収束に関する議論	60
6.2.4	圧力依存性の取り方	61
第 7 章	電子構造スペクトル	62
7.1	DOS 構造	62
7.1.1	準備	62
7.1.2	DOS 構造の結果	64
7.1.3	表示	66
7.1.4	フェルミ面	68
7.2	バンド構造	69
7.2.1	準備	69
7.2.2	バンド構造の結果	70
7.3	波動関数表示	76
第 8 章	分子動力学法	77
8.1	シミュレーションの準備	78
8.2	シミュレーションの結果	80
8.2.1	有限温度シミュレーション	83
終章		88
付 録 A	結晶の Mathematica によるグラフィックス表示	90
A.1	グラフィックス座標の取り方	90
A.2	電荷密度分布	93
謝辞		95
参考文献		96

序論

パッケージ「Osaka2002_nano」(略称「Osaka2002」)は密度汎関数理論に基づく固体の電子状態計算のプログラム集である。具体的手法としては擬ポテンシャル法による平面波展開法を用いることで、固体の電子状態計算から、構造の最適化、フォノン、光学定数の計算、さらに有限温度の分子動力学シミュレーションなど広範囲の電子状態計算を行える。

今日では「第一原理計算」と呼ばれるものは、その精度ゆえかなりの信頼で受け入れられてきている。化学の分野ではもうそのようになっていくような気がするが、固体物理の分野でも、実験家が得たデータの解釈や実験予測などに日常的に第一原理計算を活用することはそんなに遠くないように思える。

しかしながら現実には、固体物理の分野で実験家の人がそのように第一原理計算プログラムを使おうとするとかなりギャップがあるのではないだろうか？たぶん多くの人にとって一番大きな問題なのは、計算技術を取得するためのバリアであろう。

今日では「公開」と称されているプログラムはかなりの数に上ると思われるが、その多くはソースプログラムを公開しているだけという意味が多い。実際には使い方を伝授してもらわねばまず使うことはできない。簡単な使い方について書かれた書類が添付されているものもあるが、それらはあくまでそれが使用される研究室の環境を想定しているわけで、実際の使用に際しては適当な経験者の指導があって初めて使えるようになる。

今日の第一原理計算は、あたかも複雑な建築物のように様々な要素から構成されていて、各要素の理論はそれぞれに深く、そのためそれを使いこなすためかなりの訓練が必要となる。「非経験的計算」を習得するためには言葉とは裏腹に「経験」が必要ということである。そのような学習過程はとても簡単な取り扱い説明書で尽くされるものではない。

しかしながら今日の実験家に要請されていることを考えるに、この現状を放置しておくことはできない。ある程度理論的バックグラウンドに興味を覚える人であれば使えるようなプログラムを提供することが理論家に求められている。今回、我々はこのような時代的要請に応えるべく、擬ポテンシャルを使った第一原理計算プログラムパッケージ「Osaka2000」を開発し、一般に公開した。プログラム開発を行う上で念頭をおいたことは、「適当な指導者を想定せず、実験家が独習だけで習得できるプログラムを作ること」である。これは目指すべくして、なかなか到達し難い目標であることは十分承知の上で、挑戦している。

公開して2年以上の歳月を経て、我々としてもかなりユーザーの声を聞いてきてあれやこれや改良点のあることは十分に承知しているつもりである。特にプログラミングスタイルの点ではかなり問題があった。古いFortran77のスタイルでは計算するたびにコンパイルしなければならず、そのためのパラメータをインクルーズファイルに書き込むのがかなり初心者には負担になっていた。これを間違えると、コンパイルが通っても実行時に、何のメッセージも残さずコアダンプを引き起こしたりして、何が悪かったのかさっぱりわからないという苦情が寄せられることになる。

また我々開発者にとっても、プログラムのさらなる発展を目指そうとすると、古い Fortran77 の本源的な限界をつくづく感じるようになった。各プログラムはもともと統合化を意識して書かれていないため、それらを統合しようとするとなすますプログラムが複雑になり、プログラム開発者にとっても非常に読みにくいものとなった。当事者でさえ読むのが困難なものに誰が開発者として加わってくるだろうか？それがプログラム開発の大きな桎梏となることは開発の初期段階からわかっていたことである。しかしかといって初めから作り直そうとした時、それに要するであろう作業量を想像すると容易に決断できるものではなかった。しかしながらそれをいつまでも引き伸ばしていたのでは事の根本的解決にはならない。遂に決断した。2001年春のことである。それから全てのコードを全くクリーンな状態から書き下ろした。始めから書くと決意したことで、それまで様々な根本的制約を持っていた Fortran77 を捨て、f90 の洗練されたスタイルを採用することが出来た。そうして出来たバージョンが「Osaka2002_nano」である。でき上がるまでは本当に苦しかった。一つの改定はまた新たな間違いを引き起こす。開発はこの類いのバグとの戦いの連続であった。f90 で書き直して良かったと思えるようになったのはようやくつい最近のことである。

バージョンが進むにつれて、マニュアルを書くことの困難さにも直面している。プログラムが進化するごとに、このプログラムで可能となることやまたその詳細を記述することは変更を受けるし、だいたいどこまで記述するかに関する考え方はその度が変わる。そうした経験より、こうした手引書では動作の基本だけを記述すべきだと確信した。そのためまずプログラムのインストールのような技術的詳細は別にインストールマニュアルに記載することにした。また個々の応用の詳細は、応用編としてその都度別に著すことにし、それらをテクニカルレポートシリーズとして準備した。

歴史的ノート

多くのプログラムがそうであるように、このプログラムも一人の人によってのみ作成されたのではない。都合上、このマニュアルには著者一人だけの名前が出ているが、勿論先駆者の仕事があって初めてなされたものである。そこでこのプログラム群の歴史を述べることで、作者のそれらの先駆者達に対する感謝の意に代えたい。(敬称略)

まず、ポテンシャルをつくる atom プログラムは(それ自身がまた長い経歴を持っているが) Troullier と Martins がつくったものを使用している [10]。これはもともとパッケージ「Osaka2000」と独立したものである。

パッケージ「OSAKA2000」のコアとなるプログラム pwm は、Car-Parrinello 法にもとづく「第一原理分子動力学シュミレーション」プログラム cpgs を元にして、作者が発展させたものである。元となったプログラム cpgs は 1987 年から 1995 年にかけて、当時東北大において吉田博の指導下で、織田望(現電総研)、佐々木泰造(現金属材料技術研)、西松毅(現東北大金研)によって開発された。このコードは当時は半導体中の不純物を研究する目的で使われていて、その目的では既に完成品であった。従って最も困難な部分を達成した彼らこそこのプログラムの生みの親である。

但し目的がそのようなものであったため、一般の結晶にはそのまま適用できなかった。1998年より著者が、全ての結晶に対応するように、柳瀬章によって開発された空間群のプログラム「TSPACE」[24]とのドッキングをはかり、それと同時に電子系の収束に Teter-Payne-Allan のアルゴリズム [18] を取り入れたり、原子位置、格子定数に関する最適化に共役勾配法を取り入れるなどほとんどのルーチンを書き換えた。

さらにその周辺のバンド計算、フォノン計算などのプログラムを開発してきたが(その出力の作画に際しては柳瀬章らによる ayband などを利用している)、この際各プログラムの様式などを統一して一つのパッケージにまとめたものが、「OSAKA2000」である。公開したのは2000年秋のことである。

その後それを発展させたものが「Osaka2002.nano」である。

引用に際して

「Osaka2002」を使用して論文を発表される場合、本文か Acknowledgment にパッケージ「Osaka2002」を使用した旨明記していただくようお願いする。「Osaka2002」はプログラム名であるので、一般の読者がどんな計算をしているのかわからない。それで原理に関する論文の引用が必要である。どこまで引用すべきかはもちろん任意であるが、もし論文が計算を実験データの解釈に使われるくらいの類であるなら、次の2点くらいを推奨しておく。

使用したポテンシャルに関しては Troullier-Martins のものを [10]、計算方法に関しては Payne らによるレビュー論文 [19] がある。

バンド図に関しては、図 7.5 のような簡易的なものは構わないが、図 7.6 のような精巧なバンド図は、柳瀬章による ayband で初めて可能になったものである。その際、TSPACE[24] 及び同氏による ayband の恩恵にあずかったことも一言述べるべきである。

保証

この種のプログラムの配付ではいつものことながら、結果の保証について述べなければならぬ。私達はこれが正しい解を与えることに最大限の努力を払っているが、それにもかかわらず最終的な結果に対する社会的な責任は負わないことをご了承いただきたい。

ソースコード

ソースコードの配布は以下のサイトで行っている。

<http://www.cmp.sanken.osaka-u.ac.jp/~koun/osaka.html>

バグレポート及び、マニュアルの誤りは koun@sanken.osaka-u.ac.jp まで御願います。

第1章 理論

初めに、第一原理に基づく擬ポテンシャル法の概略を説明する。¹ 理論の核心部分は他の教科書で勉強の方がより適切であるが、特に第 1.5 節から 1.7 節の事項は、比較的計算上の技術的なことに属し(しかしプログラム `pwm` を動かすうえで重要である)教科書ではあまり書かれていないので、少し詳細に述べる。

1.1 密度汎関数法

理論上、固体の性質はその系の完全なハミルトニアン²の固有状態を解くことによって得られる。非相対論的ハミルトニアンは原子単位系で、

$$H = - \sum_i \nabla^2 + \frac{1}{2} \sum_{i \neq j} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_i V_{\text{ion}}(\mathbf{r}_i) + E_{\text{ion}}(\{\mathbf{R}_n\}) \quad (1.1)$$

と表される²。ここで、 i, j の和は系の全電子について行われる。 \mathbf{R}_n は原子核の位置、 $V_{\text{ion}}(\mathbf{r}_i)$ は原子核によるポテンシャル、 $E_{\text{ion}}(\{\mathbf{R}_n\})$ は原子核間の相互作用エネルギーである。定常状態の全エネルギーや、原子間力、電子密度、静電ポテンシャル等のような実際に興味の対象となる多くの性質は、定常状態の波動関数に対する Schrödinger 方程式、

$$H\Psi_0(\{\mathbf{r}_i\}) = E_0\Psi_0(\{\mathbf{r}_i\}) \quad (1.2)$$

を解くことによって得られる。固体の性質の非経験的解析の試みのひとつは、多電子系のハミルトニアン、式 (1.1) に対する方程式を、直接解くことである。その際、多数の一電子波動関数の組み合わせになるスレーター行列一つで式 (1.2) を展開して解くハートリー・フォック近似が一般的である。これは交換効果のみを考慮したものになるが、それを超える計算には、多数のスレーター行列を用いた展開による配置間相互作用法 (configuration interaction) や量子モンテカルロ法などが使われる。いずれにしても、これらは波動関数を基本に置いて、固体の電子状態を多数の波動関数の組み合わせで表そうとするものである。配置間相互作用法の方法では特にこの波動関数の組み合わせが膨大複雑になるため、計算負荷が大きく、扱える問題のサイズに制限が生じる。

¹それにしても「第一原理計算」と称するものが如何に多いことか。作者は「第一原理」というからには本来それだけあれば他に何もいらぬことを意味すると思うのだが、現実には、擬ポテンシャルあり、LAPW、KKR 法あり、さらに化学の分野ではハートリー・フォックや $X\alpha$ 法なども「第一原理計算」と称している。要するに非経験的電子計算を総称して「第一原理計算」というのだろう。このところは英語でははっきり First-principles calculations と「原理」が複数あることを認めている。もっともそれが、上に意味で方法が多数あるという意味なのか、それとも擬ポテンシャルに限ってもそれがこれから述べるいくつかのパーツからなるという意味なのだろうか。

²原子単位では、 $e^2 = 2$ とするが、ここでは e^2 はそのまま残した。原子単位と言っても、Rydberg 単位と、Hartree 単位があるので、混乱を避けるためである。

一方、多体問題に対するこれとは全く異なるアプローチとして、電子密度を第一義的な変量として考える密度汎関数法 (Density Functional Theory) が出現した。セルフコンシステントな一電子方程式に移し替えられた密度汎関数法の方程式を解くことは、式 (1.2) を解くよりもはるかに簡単で、相関効果を含んでおり、計算可能な系のサイズもはるかに大きい。1980年頃以来、この方法は、固体や分子の性質を第一原理から計算する主な方法のひとつとしての地位を確立してきた。

密度汎関数理論の元となる論文は Hohenberg & Kohn [1] である。この論文では、外部ポテンシャル中の電子ガスの基底状態エネルギーは電子密度の一意的な汎関数になる、ということが示されている。そのすぐ後に Kohn と Sham [2] は、式 (1.2) の基底状態を解くことは、汎関数 $E_{\text{KS}}(\{\Psi_i\})$ を一電子波動関数 $\Psi_i(r)$ についての規格直交化条件のもとで最小化することと同じであることを示した。

$$E_{\text{KS}}(\{\Psi_i\}) = - \sum_i f_i \int \Psi_i \nabla^2 \Psi_i d^3\mathbf{r} + \int \rho(\mathbf{r}) V_{\text{ion}}(\mathbf{r}) d^3\mathbf{r} + \frac{e^2}{2} \int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d^3\mathbf{r} d^3\mathbf{r}' + E_{\text{xc}}[\rho(\mathbf{r})] + E_{\text{ion}}(\{\mathbf{R}_n\}) \quad (1.3)$$

式 (1.3) において、 E_{KS} は Kohn-Sham の汎関数で、添字 i の和はすべての一電子軌道について行われ、 f_i は状態 i での占有数、 E_{xc} は相関交換効果の汎関数である。 $\rho(\mathbf{r})$ は電荷密度であり、以下の式で与えられる。

$$\rho(\mathbf{r}) = \sum_i f_i |\Psi_i(\mathbf{r})|^2 \quad (1.4)$$

式 (1.4) の Kohn-Sham 汎関数を最小化する波動関数 $\Psi_i(\mathbf{r})$ は以下の固有値方程式に従う。

$$H_{\text{KS}}\Psi_i = \epsilon_i\Psi_i \quad (1.5)$$

ここで、 H_{KS} は Kohn-Sham のハミルトニアンであり、

$$H_{\text{KS}} = -\nabla^2 + V_{\text{ion}}(\mathbf{r}) + V_{\text{H}}(\mathbf{r}) + V_{\text{xc}}(\mathbf{r}) \quad (1.6)$$

$V_{\text{H}}(\mathbf{r})$ はハートリーポテンシャルで、

$$V_{\text{H}}(\mathbf{r}) = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d^3\mathbf{r}' \quad (1.7)$$

と表され、 V_{xc} は交換相関ポテンシャルである。

$$V_{\text{xc}}(\mathbf{r}) = \frac{\delta E_{\text{xc}}[\rho]}{\delta \rho(\mathbf{r})} \quad (1.8)$$

ϵ_i, Ψ_i はそれぞれ、Kohn-Sham 方程式の固有値、波動関数を表す。

式 (1.5) で計算される波動関数は、ハートリーポテンシャルや交換相関ポテンシャルをつくる電荷密度と同じ電荷密度を作り出さなければならない、という意味で、Kohn-Sham 方程式はセルフコンシステントである。

式 (1.6) は一見、それまでの一電子波動関数に関する波動方程式と代わり映えしないように見えるが、そこに込められている思想が違うのである。ハートリー・フォックの場合

は、波動関数が第一義的量で、密度はそれから導かれる二次的量とみなされる。一方密度汎関数理論では、密度こそ第一義量で、波動関数の方は、密度を出すための一種の方便で、同じ密度を与えるものならどのように取ってもよい。

密度汎関数理論、及び次節で述べられる局所密度近似についてはたくさんの良いレビューがある（文献 [3, 4, 5]）。

1.2 局所密度近似

多電子問題を一電子問題に置き換えることによる密度汎関数法の数学的単純化は、一方では未知の電荷密度の汎関数、すなわち交換相関エネルギーを導入しなければならないという厄介さを合わせ持つ。幸い、 E_{xc} についての簡単な近似が存在する。最も一般的な E_{xc} の形は局所密度近似 (The Local Density Approximation, LDA) を使うことで得られる。すなわち、知られている自由電子ガスに対する交換相関エネルギーを利用する。この近似では、系の空間の各点での交換相関エネルギーは、その電荷密度での一様な自由電子ガスの交換相関エネルギーに等しい、とされる。

$$E_{xc} = \int \epsilon_{xc}(\mathbf{r})\rho(\mathbf{r})d^3\mathbf{r} \quad (1.9)$$

交換相関ポテンシャル $V_{xc}(\mathbf{r})$ は、

$$V_{xc}(\mathbf{r}) = \frac{\delta E_{xc}}{\delta \rho(\mathbf{r})} = \frac{\partial \{\rho(\mathbf{r})\epsilon_{xc}(\mathbf{r})\}}{\partial \rho(\mathbf{r})} \quad (1.10)$$

そして、 $\epsilon_{xc}(\mathbf{r})$ は

$$\epsilon_{xc}(\mathbf{r}) = \epsilon_{xc}^{\text{hom}}[\rho(\mathbf{r})] \quad (1.11)$$

で、 $\epsilon_{xc}^{\text{hom}}$ は一様な電子ガスにおける交換相関エネルギーである。局所密度近似の範囲内で、交換相関ポテンシャルは電荷密度の局所的な関数となる。この近似の有効性と正確さは、固体や分子などの系での多くの計算結果が証明してくれている。ここでは、密度の関数としての交換相関エネルギーは量子モンテカルロ法 [6] から得られる結果をパラメータ化したものを用いている [7]。

1.3 擬ポテンシャル近似

局所密度近似に続く第二の近似は擬ポテンシャルの使用 (The Pseudopotential Approximation) である。後で述べられるが、固体の波動関数はここでは平面波で展開される。平面波展開は局在化された状態を表すのに非常に不経済な表現である。コアの部分で激しく振動している内殻状態を記述するには適切ではない。幸い、物質の多くの物理的、化学的性質は主に広がった価電子状態により決まり、原子核付近に局在している内殻状態には鈍感である。そこで物質の結合状態などを記述する上で、価電子状態だけで近似し、それゆえポテンシャルとしては、急激に変化する元のポテンシャルに比べて遥に緩やかな変化を持つポテンシャルで価電子状態をシュミレーションしようという考えが生まれた。これが擬ポテンシャル法である [8]。それに伴う価電子状態をシュミレートする波動関数は擬波動関数と呼ばれる。

擬ポテンシャルは、化学的に重要である原子の外側領域において全電子原子の電子散乱特性にできるだけ似せて作られる。このように作られた良い擬波動関数は「transferableである」と言われる。一般的に、擬ポテンシャルはそれぞれの角運動量成分で異なる散乱ポテンシャルを持ち、かつ非局所的である。すなわち擬ポテンシャルはオペレータとして、

$$\hat{V}_{NL} = \sum_l |lm\rangle V_l \langle lm| \quad (1.12)$$

ここで $|lm\rangle$ は球面調和関数で、 l, m はそれぞれ角運動量、方位角運動量である。全電子ポテンシャルは局所ポテンシャルである。真の波動関数と擬波動関数はコア半径の外側で一致するため、擬ポテンシャルの非局所性は原子コア領域内に限られる。実際には、軽い原子では擬ポテンシャルに $l = 2$ 以上の非局所項を加えることは必ずしも必要ではない。

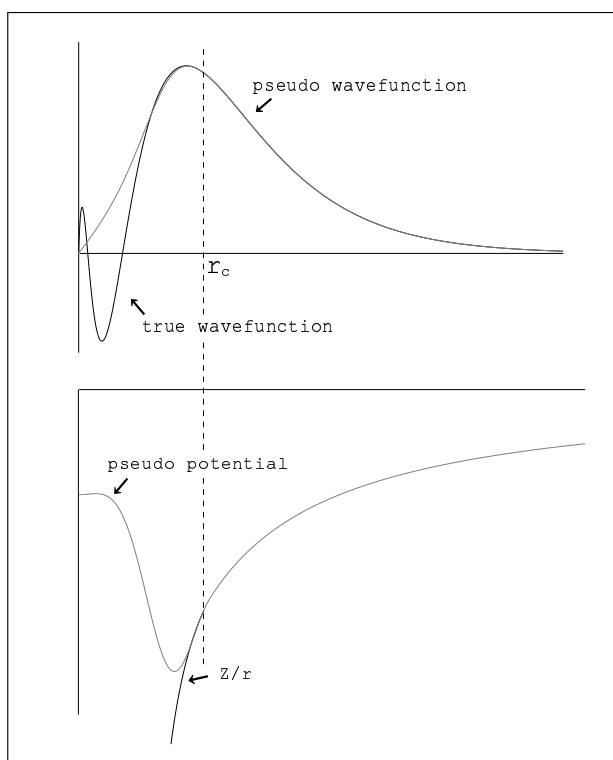


図 1.1: 擬ポテンシャルの概略 振動する真の波動関数 (実線) は節のない擬波動関数 (破線) に置き換えられる。全電子、擬原子それぞれの波動関数、およびポテンシャルはコア半径 r_c より外側の化学的に重要な領域で一致する。

擬ポテンシャルを生成するため、過去多くの方法が編み出されてきたが [8]、重要な進歩は Hamann らによる波動関数の「ノルム保存」という概念でもたらされた [9]。これにより擬ポテンシャル法が固体の問題を解く方法として、今日では最も一般的に用いられるようになったと言って過言ではないだろう。このアイデアでは、節のない擬波動関数はコア半径 r_c より外では真の波動関数と一致するように取られる。その他いくつかの条件を加えて、擬ポテンシャルが作られるのであるが、その様子を図 1.1 に示す。滑らかな擬ポテンシャルが得られていることがわかる。

ここでのプログラムでは、擬ポテンシャルとして、「ノルム保存」型で、より計算効率を改良した Troullier-Martins 型を採用している [10]。

式 (1.12) で定義された擬ポテンシャルは semi-local と呼ばれることがある。位置の角度成分に関しては非局所的であるが、動径方向成分に関しては局所的であるからである。それに対して、Kleinman、Bylander は動径方向成分も非局所的に扱う full-nonlocal 型を提唱している [11]。それによると、

$$V_{\text{ion}} = V_L + \sum_{lm} \frac{|\Phi_{lm}^0 \Delta V_l\rangle \langle \Delta V_l \Phi_{lm}^0|}{\langle \Phi_{lm}^0 | \Delta V_l | \Phi_{lm}^0 \rangle} \quad (1.13)$$

ここに Φ_{lm}^0 は擬ポテンシャルを作るときの原子の擬波動関数で、

$$\Delta V_l = V_{l,\text{NL}} - V_L \quad (1.14)$$

となる。これにより、非局所ポテンシャルの計算が早くなるだけでなく、 V_L の任意性を利用すると、非局所ポテンシャル行列と波動関数の積の計算も加速される。

1.4 平面波展開

無限に続くマクロな結晶では $\sim 10^{23}$ 個の原子がある。このようなほとんど無限次元とも言える自由度を持つ系に対してまともに KS 方程式を解くことは適切でない。そこで編み出されたのが、Born-von Karman の境界条件という人為的だが非常に便利な数学的道具である。それに付随してもう一つ重要な数学的定理が Bloch の定理である [13]。

平面波法に限らず、全ての固体の計算法はこの周期境界条件の恩恵にあずかっている。格子欠陥を持つ結晶や表面、さらにアモルファスなど周期性のない固体の場合、もちろん基本単位格子というものは存在しないが、人為的な巨大な単位格子を取るスーパーセル法によって計算可能となる。

結晶の数学的モデルは、実空間での三つの基本並進ベクトルで作られる。すなわち、

$$\mathbf{t} = n_1 \mathbf{R}_1 + n_2 \mathbf{R}_2 + n_3 \mathbf{R}_3 \quad (1.15)$$

で与えられる。 n_1, n_2, n_3 は任意整数である。このように基本並進ベクトルで作られる格子をブラベー格子という。結晶はブラベー格子と単位構造 (basis) の組み合わせで表現される。

この格子に対して、逆格子空間が定義される。逆格子空間の基本並進ベクトルは

$$\mathbf{R}_i \cdot \mathbf{G}_j = 2\pi \delta_{ij} \quad (1.16)$$

を満たすように取られる。 i, j はそれぞれ 1 から 3 をとる。 \mathbf{G}_1 は以下ようになる。

$$\mathbf{G}_1 = 2\pi \frac{\mathbf{R}_2 \times \mathbf{R}_3}{\mathbf{R}_1 \cdot (\mathbf{R}_2 \times \mathbf{R}_3)} \quad (1.17)$$

$\mathbf{G}_2, \mathbf{G}_3$ も同様の形で与えられる。逆格子空間での格子ベクトル (逆格子ベクトル) は次式のように与えられ、ここで n_1, n_2, n_3 は任意整数である。

$$\mathbf{G} = n_1 \mathbf{G}_1 + n_2 \mathbf{G}_2 + n_3 \mathbf{G}_3 \quad (1.18)$$

逆格子ベクトルは任意の周期関数のフーリエ展開に使われる。 $f(\mathbf{r})$ がブラベー格子ベクトル $\{\mathbf{R}\}$ について滑らかな関数であるとき、

$$f(\mathbf{r}) = \sum_{\mathbf{G}} A_{\mathbf{G}} e^{i\mathbf{G}\cdot\mathbf{r}} \quad (1.19)$$

となる。 $A_{\mathbf{G}}$ は逆格子ベクトル \mathbf{G} に関するフーリエ展開係数である。

周期境界条件の下では電子状態は波数ベクトル \mathbf{k} とバンド指数 n を用いて指定される。そして波動関数は基本単位格子の周期関数 $u_{\mathbf{k}}(\mathbf{r})$ に $\exp(i\mathbf{k}\cdot\mathbf{r})$ を掛けた形をしている。すなわち、

$$\Psi_{\mathbf{k}n}(\mathbf{r}) = \exp(i\mathbf{k}\cdot\mathbf{r})u_{\mathbf{k}}(\mathbf{r}) \quad (1.20)$$

と表される。³ これがブロッホの定理の主張するところであるが、このおかげで解くべき問題の自由度がほとんど無限大から、単位格子内の原子数のオーダーくらいに減少できるのである。現実の多種のバンド計算は(セル法を除いて)、式(1.20)のブロッホ関数を表す基底関数の取り方が違うと見ることができる。

式(1.20)の中で、 $u_{\mathbf{k}}(\mathbf{r})$ は格子の周期関数であるから、式(1.19)に従って逆格子ベクトルで展開できる。つまり、

$$\Psi_{\mathbf{k}n}(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{k}+\mathbf{G}} e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \quad (1.21)$$

従って平面波法はそのままブロッホの条件を満たし、かつ空間的に片寄りのない電荷密度をもたらせるので、ブロッホ関数の基底関数としては自然である。そのため全エネルギーなどの物理量の表式が一筋で書けるという利点がある。最大の欠点は収束が悪いということである。(現実的なポテンシャルで如何に平面波展開が効率の悪いものであるかの例は、例えば文献[8]の第2章で挙げられている。)

平面波で展開された、全エネルギーの表式(1.3)は

$$\begin{aligned} E_{\text{tot}} = & \sum_{i,\mathbf{G}} |c_{\mathbf{k}_i+\mathbf{G}}|^2 (\mathbf{k}_i + \mathbf{G})^2 + \frac{1}{2} \sum_{\mathbf{G}} \rho^*(\mathbf{G}) V_{\text{H}}(\mathbf{G}) + \frac{3}{4} \sum_{\mathbf{G}} \rho^*(\mathbf{G}) V_{\text{xc}}(\mathbf{G}) \\ & + \sum_{\mathbf{G}} \rho^*(\mathbf{G}) S^*(\mathbf{G}) V_{\text{L}}(\mathbf{G}) + \sum_{i,l,\mathbf{G},\mathbf{G}'} c_{\mathbf{k}_i+\mathbf{G}}^* c_{\mathbf{k}_i+\mathbf{G}'} \\ & \times S(\mathbf{G}' - \mathbf{G}) V_l^{\text{NL}}(\mathbf{k}_i + \mathbf{G}, \mathbf{k}_i + \mathbf{G}') + E_{\text{ion}}(\{\mathbf{R}_n\}) \end{aligned} \quad (1.22)$$

となる[14]。 $S(\mathbf{G})$ は構造因子である。さらに後に述べる Hellmann-Feynman 力やストレスなどもこの平面波展開により、評価しやすい形になる。

式(1.22)の最後の項は、イオン-イオン間直接クーロン相互作用で、Ewald 和と呼ばれているものである。それは

$$E_{\text{ion}}(\{\mathbf{R}_n\}) = \frac{e^2}{2} \sum_{\kappa,\kappa'} Z_{\kappa} Z_{\kappa'} \gamma_{\kappa,\kappa'} \quad (1.23)$$

³指数部を $\exp(-i\mathbf{k}\cdot\mathbf{r})$ と取ることもできる。このプログラムでは TSPACE を利用するため、式(1.20)のように指数部は正と取り決めている。

で、 κ は単位格子中の原子のインデックスで、 $\gamma_{\kappa,\kappa'}$ は

$$\begin{aligned} \gamma_{\kappa,\kappa'} = & \sum_{\nu'} \frac{\text{erfc}(\eta |\mathbf{R}(\nu'_{\kappa'} - \nu_{\kappa})|)}{\mathbf{R}(\nu'_{\kappa'} - \nu_{\kappa})} \\ & + \frac{4\pi}{\Omega_c} \sum_{\mathbf{G} \neq 0} \frac{1}{G^2} \exp\left[-\left(\frac{G}{2\eta}\right)^2\right] \exp[i\mathbf{G} \cdot (\mathbf{x}_{\kappa} - \mathbf{x}_{\kappa'})] \\ & - \frac{2\eta}{\sqrt{\pi}} \delta_{\kappa,\kappa'} - \frac{\pi}{2\eta^2 \Omega_c} \end{aligned} \quad (1.23a)$$

で与えられる。プライムのついてる和は同じ原子を除くことを意味している。 η は式 (1.23a) で実空間にわたる和と、逆格子空間にわたる和での収束が良くなるように取る (例えば文献 [15] p. 385)。

1.5 平面波のカットオフ

式 (1.21) は無限の逆格子ベクトル \mathbf{G} についての和である。もちろん現実にはどこかで切らなければならない。実際の計算では、あるカットオフエネルギー E_{cut} (原子単位) に対して、それよりも小さいエネルギーの平面波のみが選ばれる。すなわち

$$|\mathbf{k} + \mathbf{G}|^2 < E_{\text{cut}} \quad (1.24)$$

これは、逆格子空間の球で定義され、球の半径 (カットオフ半径 k_c) よりも小さな波数ベクトルの平面波が全て含まれることを意味する。図 1.2 参照。このときの平面波カットオ

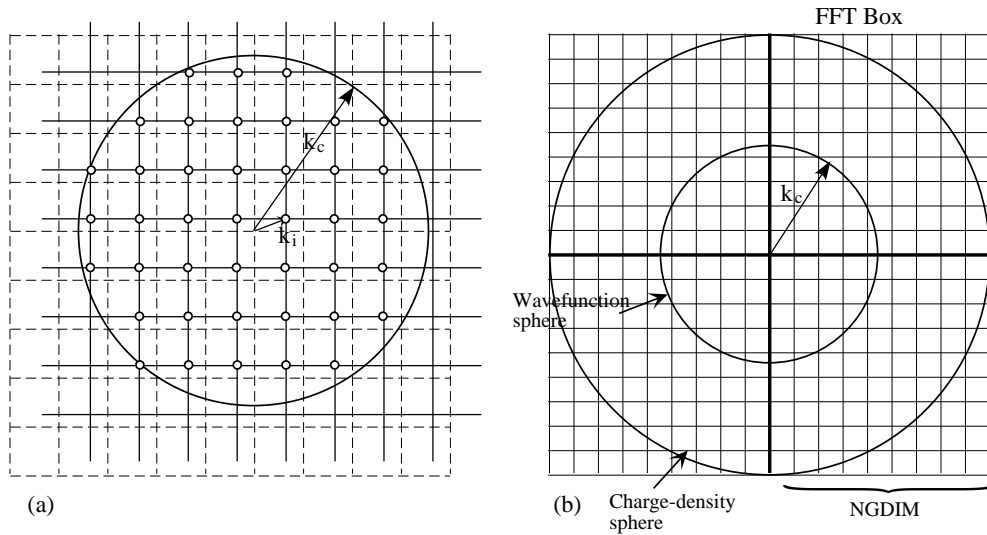


図 1.2: a) 点 \mathbf{k}_i に対する平面波の取り方。b) 平面波球、電荷密度球、FFT の矩形の関係。

フ半径は

$$E_{\text{cut}} = k_c^2 \quad (1.25)$$

で与えられる。

カットオフ半径 k_c 内の平面波の数 N_{pw} は

$$N_{\text{pw}} = \frac{\frac{4\pi}{3}k_c^3}{\Omega_c} \quad (1.26)$$

と見積られる。ここに Ω_c は基本単位格子の体積である。

計算の収束に必要とされる平面波の数 N_{pw} は基本単位格子内の原子の種類に依存する。周期表で一列目や、一列目の遷移金属を含む系は一般的に、二列目や三列目だけからなる系よりも大きな平面波のカットオフエネルギーを必要とする。

1.6 k 空間上の総和

全エネルギーや力の計算では、固体の Kohn-Sham 波動関数に作用するオペレータ \hat{Q} についての平均を知ることがしばしば必要とされる。ブロッホの定理によれば、全波動関数に対する平均は、

$$Q = \frac{\Omega_c}{(2\pi)^3} \int_{\text{BZ}} Q_{\mathbf{k}} d^3\mathbf{k} \quad (1.27)$$

ここで $Q_{\mathbf{k}}$ は、

$$Q_{\mathbf{k}} = \langle \Psi_{\mathbf{k}n} | \hat{Q} | \Psi_{\mathbf{k}n} \rangle \quad (1.28)$$

と表される。例えば、 \hat{Q} が Kohn-Sham ハミルトニアンであるような場合、 $Q_{\mathbf{k}n}$ は Kohn-Sham の固有値で、 Q はバンド n でのバンドエネルギーの平均値になる。

数値計算では式 (1.27) の積分はもちろん有限メッシュ上の和に置き換えられるが、通常はそれに膨大な時間がかかる。しかしながら、k 空間のメッシュ点である特別の点のセットを選ぶと、少数の点で多数のサンプリングをしたのと同じ精度を得ることができる。これが特別点サンプリング法と呼ばれるものである。

本プログラムでは標準的に、特別点サンプリング法として Monkhorst と Pack に依るもの [16] を用いている。k 点のメッシュは N_1, N_2, N_3 の三つの整数を用いてつくられる。これらの整数は基本単位逆格子での k 点の密度を決定する。メッシュの一般点は次式で与えられる。

$$\mathbf{k}_{rst} = u_{1r}\mathbf{G}_1 + u_{2s}\mathbf{G}_2 + u_{3t}\mathbf{G}_3 \quad (1.29)$$

$$u_{ip} = \frac{2p - N_i - 1}{2N_i} \quad (1.30)$$

p は 1 から N_i までをとる。このメッシュはブリルアン・ゾーンでの $N_1N_2N_3$ 個の k 点を作る。したがって、式 (1.27) の積分は不連続な k 点の和に置き換えられる。

$$Q = \frac{1}{N_1N_2N_3} \sum_{rst} Q_{\mathbf{k}_{rst}} \quad (1.31)$$

実際には、いくつかの k 点は結晶の時間反転操作かあるいは点群操作によって互いに対称になる。もし複数の k 点に対称ならば、Kohn-Sham 方程式は対称な点のうち的一点でのみ解けばよく、残りは対称操作から導くことができる。この対称性のおかげで計算時間は

大幅に削減される。プログラム中では自動的に結晶の空間群についての情報を用いて、計算に使う k 点の数を可能な限り減らしている。

k 点の特別点サンプリングがどれくらい良いものであるか、その質は実空間のカットオフベクトルの中で見積もられる。 Q_k のような逆格子空間の周期関数は実空間のフーリエ級数表現を持つ。

$$Q_k = \sum_{\mathbf{R}} B_{\mathbf{R}} e^{i\mathbf{k} \cdot \mathbf{R}} \quad (1.32)$$

ここで $B_{\mathbf{R}}$ は展開係数である。ゾーン全体にわたる Q_{kn} の平均は B_0 である。通常はフーリエ係数 $B_{\mathbf{R}}$ は $|\mathbf{R}|$ の増加とともに急激に減少する。(1.32) を (1.31) に代入すると、

$$Q = B_0 + \sum_{\mathbf{R} \neq 0} B_{\mathbf{R}} \phi_{\mathbf{R}} \quad (1.33)$$

$\phi_{\mathbf{R}}$ は

$$\phi_{\mathbf{R}} = \frac{1}{N_1 N_2 N_3} \sum_{rst} e^{i\mathbf{k}_{rst} \cdot \mathbf{R}} \quad (1.33a)$$

と書け、サンプリング k_i のセット $\{rst\}$ に対する $\mathbf{R} \neq 0$ の和は Q の誤差とみなせる。 \mathbf{R} をその大きさ $|\mathbf{R}|$ で分類すると、 \mathbf{R} の殻構造ができるが、その殻 i ごとに式 (1.33) を評価し、

$$\sum_{\mathbf{R}}^{\text{ith shell}} \phi_{\mathbf{R}} \quad (1.33b)$$

ができるだけ遠くの殻まで 0 となれば、そのサンプリングの質は良いということになる。

1.7 Kohn-Sham 汎関数における共役勾配最小法

次に、式 (1.3) を解かなければならない。従来は式 (1.5) を対角化し、出力電荷密度と、入力電荷密度が同じになるまで繰り返し、セルフコンシステント解を得ていた。この方法だと、出力電荷密度と入力電荷密度を混ぜ合わせる割合が微妙で、特に扱う形のサイズが大きくなると、収束しなくなることがしばしばあった。Kohn-Sham 汎関数における共役勾配最小法はこの難点を解決した優れた方法であるので、ここで詳しく説明する。

これまでの議論より、系の基底状態は Kohn-Sham のエネルギー汎関数 $E_{KS}(\{c_{\mathbf{k}+\mathbf{G},n}\})$ 関数を占有された波動関数の平面波展開係数 $\{c_{\mathbf{k}+\mathbf{G},n}\}$ で、最小化させる問題と考えることができる。

多変量の関数 $f(\mathbf{x})$ の極小値を求める数学的技法として、共役勾配法 (Conjugate-Gradient Method) が有効であることが知られている [17]。これは特に $f(\mathbf{x})$ が

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \cdot \mathbf{x} + c \quad (1.34)$$

と線形近似でき、かつ \mathbf{A} が正値行列 (positive-definitive) のとき必ず最小値を見いだすのに成功する (少なくとも原理的には)。

関数 $f(\mathbf{x})$ の点 \mathbf{x} での勾配は

$$-\nabla f(\mathbf{x}) = \mathbf{b} - \mathbf{A} \cdot \mathbf{x} \quad (1.35)$$

であるので、 $f(x)$ が極小値をとるときの x は線形方程式

$$A \cdot x = b \quad (1.36)$$

を満たす。

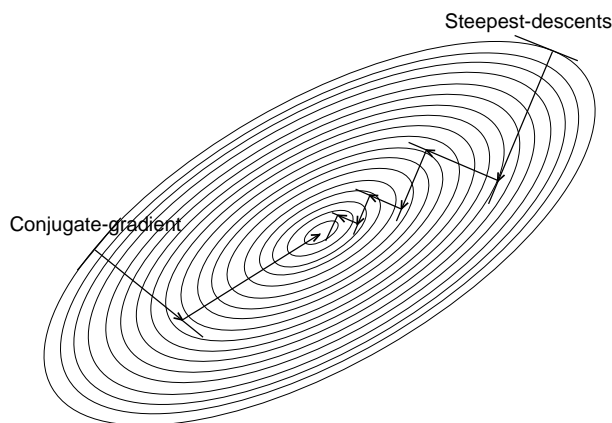


図 1.3: 最急勾配法と共役勾配法の比較。

この極小化問題に対する原始的な解法は、まず試行点 x_0 で式の (1.35) を評価し勾配を求め、次にその勾配の方向に次元最小化を行う。その線上の最小点 x_1 が定まったら、次にその点で再び勾配を求め、またその方向に最小化する。という過程を繰り返すことである。これが最急勾配法 (steepest descent method) である。これは直感的に分かりやすい方法であるが、実は図 1.3 に示されるようにポテンシャルの谷間では何度もステップを繰り返すことになり非常に効率が悪い。

この最急勾配法の欠点を良く解析すると、同じ探索方向を何度も繰り返している点が問題である。過去探索した方向は二度と繰り返さない方法がある。それが共役勾配法である。

それによると、 i 番目のステップで、探索方向 h_i 上で最小点 x_i が見つかったとする。次にこの点で勾配 g_i を求める。 x_i を始点として、新しい $i+1$ 番目のステップ探索方向 h_{i+1} は

$$h_{i+1} = g_i + \gamma_i h_i \quad (1.37)$$

と前の探索方向と、現在の勾配との混ぜ合わせとなる。混ぜ合わせる割合は

$$\gamma_i = \frac{g_{i+1} \cdot g_{i+1}}{g_i \cdot g_i} \quad (1.38)$$

で決まる。これが共役勾配と言われるものであり、この方向で探索すると、図 1.3 に示されるように大変効率良く極小点が見つかる。

またこの方法では、行列 A の完全な形をする必要はなく、 A は常にベクトル x との積の形で現れているので、 $A \cdot x$ の部分がうまく評価できると勾配の計算が速くなるだけでなくメモリの節約にもなる。

密度汎関数理論により、エネルギー汎関数は真の基底状態電荷密度の時最小値を取り、それ以外の密度分布の時は必ず関数値は大きくなることを保証しているの、上の共役勾

配法は Kohn-Sham エネルギー汎関数を最小化する方法としてうってつけである。しかしながら、波動関数は規格化されていてその絶対値に関する拘束がつくこと、並びに同じ k 点の波動関数同士が直交していなければならないという拘束がつくため、事情が複雑となる。こうした困難を克服し、より効果的な共役勾配法が Teter, Payne, Allan (TPA) によって提案された [18, 19]。本プログラムはこの TPA のアルゴリズムを踏襲している。

このアルゴリズムでは、各 k 点ごとに、占有されたバンドに対して、直交規格化された初期の波動関数のセットを用意する。ここでは乱数から作られる。Kohn-Sham エネルギー汎関数はバンドごとに最小化過程を繰り返される。各 k 点の一つのバンドの波動関数の平面波展開係数に関して共役勾配最小化が適用される。その間他のバンドは固定される。一つのバンドに対して数回の共役勾配過程がなされ、そして次のバンド、次の k 点に移り、全体をさらに何回か繰り返す。

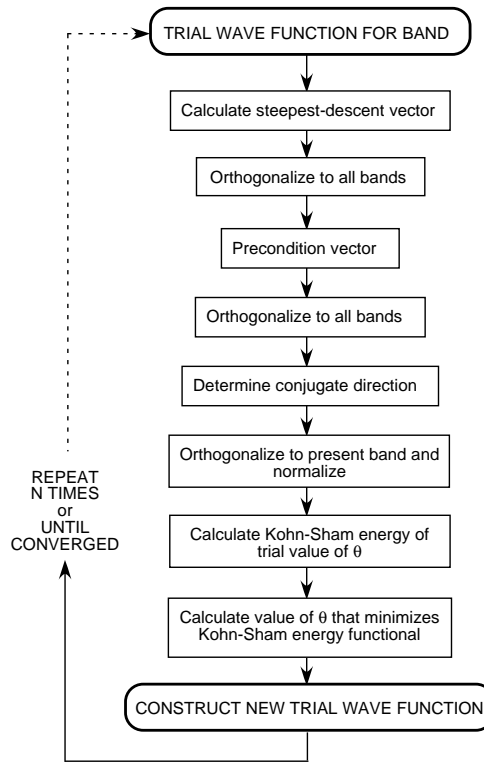


図 1.4: 直接最小化法のフローダイアグラム。

以下一つのバンド n に対する共役勾配過程を図 1.4 を参照しながら、もう少し詳しく説明する。

1. バンド n の波動関数 $\{c_{\mathbf{k}+\mathbf{G},n}\} = C_n$ に関する Kohn-Sham のエネルギー汎関数 E_{KS} の偏微分が、式 (1.6) の一電子に関するハミルトニアン H を与える。 m 番目のステップでのバンド n の波動関数 C_n^m に関する期待値は

$$\lambda_n^m = C_n^{m*} \cdot \mathbf{H} \cdot C_n^m \quad (1.39)$$

で、これによりこのハミルトニアンに対する残留誤差 \mathbf{R}^m 、すなわち最急勾配の方向が定まる。

$$\mathbf{R}^m = -(\mathbf{H} \cdot \mathbf{C}_n^m - \lambda_n^m \mathbf{C}_n^m) \quad (1.40)$$

この \mathbf{R}^m の絶対値自乗をもって波動関数の残留誤差 ζ^m と称している。

2. この残留誤差ベクトル \mathbf{R}^m が、 k 点内の他のバンドと直交するように修正する。すなわち、

$$\mathbf{R}'^m = \mathbf{R}^m - \sum_{r \neq n} (\mathbf{C}_r \cdot \mathbf{R}^m) \mathbf{C}_r \quad (1.41)$$

3. 収束を速めるため、この修正された残留誤差ベクトル \mathbf{R}'^m に対して前処理 (preconditioning) をする。すなわち適当な行列 \mathbf{K} を掛け、 \mathbf{R}'^m の各成分の大きさをおおまかに揃える。

$$\mathbf{R}''^m = \mathbf{R}'^m \cdot \mathbf{K} \quad (1.42)$$

行列 \mathbf{K} の取り方として、対角行列を持ってきて、だいたい \mathbf{R}^m の逆数になるようにすればよい。

4. \mathbf{R}''^m を改めて、自分自身を含む全てのバンドに対して直交するよう修正する。

$$\mathbf{G}^m = \mathbf{R}''^m - \sum_{r \neq n} (\mathbf{C}_r \cdot \mathbf{R}''^m) \mathbf{C}_r - (\mathbf{C}_n^m \cdot \mathbf{R}''^m) \mathbf{C}_n^m \quad (1.43)$$

5. この \mathbf{G}^m を使って、共役勾配を求める。つまり

$$\mathbf{F}^m = \mathbf{G}^m - \gamma^m \mathbf{F}^{m-1} \quad (1.44)$$

が求める共役勾配である。混ざりの係数 γ^m は式 (1.38) に従って \mathbf{G}^m から求めることができる。初期条件として $\gamma^0 = 1$ と取れば良い。

6. 再び \mathbf{F}^m を現在のバンドと直交化させかつ規格化する。それを \mathbf{D}^m とする。こうすることで、 \mathbf{D}^m 方向の一次元最小化探索が、波動関数

$$\mathbf{C}_n^{m+1} = \mathbf{C}_n^m \cos \theta + \mathbf{D}^m \sin \theta \quad (1.45)$$

に対する θ を変数としたときの全エネルギー $E_{\text{KS}}(\theta)$ 最小化問題に帰する。

7. $E_{\text{KS}}(\theta)$ は密度の関数で、密度は波動関数の自乗で与えられるので、もし考えている変数範囲で、 $E_{\text{KS}}(\theta)$ が密度に線形で近似できるならば、良い精度で、

$$E_{\text{KS}}(\theta) = \text{const} + A \cos 2\theta + B \sin 2\theta \quad (1.46)$$

とできるであろう。

式 (1.46) には 3 つの未知数が含まれているので、3 つの方程式が必要である。既に $\theta = 0$ での値 $E_{\text{KS}}(0)$ は評価してあるし、またその微分も容易に得られる。すなわち、

$$\left. \frac{\partial E_{\text{KS}}}{\partial \theta} \right|_{\theta=0} = 2f_n \text{Re}(\mathbf{D}^{m*} \cdot \mathbf{H} \cdot \mathbf{C}_n^m) \quad (1.47)$$

である。 $\mathbf{H} \cdot \mathbf{C}_n^m$ はすでに評価されているので、式 (1.47) は単にベクトルの内積を取ることで得られる。式 (1.46) の B は式 (1.47) の半分の値である。

8. 残りの一つは、 $E_{\text{KS}}(\theta)$ を 0 でない他の点 θ_1 で評価するか、あるいは $\theta = 0$ での $E_{\text{KS}}(\theta)$ の 2 次微分を求めることで得れる。ここでは後者を使っている。これにより、 A は $A = (1/4)\partial^2 E_{\text{KS}}/\partial\theta^2$ で与えられる。これらより、式 (1.46) を最小にする θ_{\min} は

$$\theta_{\min} = -\frac{1}{2} \tan^{-1} \left[-\frac{\frac{\partial E_{\text{KS}}}{\partial\theta} \Big|_{\theta=0}}{\frac{1}{2} \frac{\partial^2 E_{\text{KS}}}{\partial\theta^2} \Big|_{\theta=0}} \right] \quad (1.48)$$

で決定される。

1.8 高速フーリエ変換

共役勾配法における重要な計算部分は Kohn-Sham ハミルトニアンを波動関数にオペレートするときである。プログラムではこのステップは高速フーリエ変換 (FFT) を使って実行される。そこでは、運動エネルギーと局所ポテンシャルが逆空間、実空間のそれぞれで対角化されている、という事実を利用している。一つのバンドでの基底関数の総数が N_{pw} であると、式 (1.22) の形から、全係数において逆格子空間での運動エネルギーを計算するには $\mathcal{O}(N_{\text{pw}})$ の演算を要求することがわかる。

一方、逆格子空間でのローカルポテンシャルを用いた計算は $\mathcal{O}(N_{\text{pw}}^2)$ の演算を必要とする。なぜならばローカルポテンシャルは完全な行列だからである。ここではローカルポテンシャルの演算を書き換えることによってこの演算を加速化する。

$$\sum_{\mathbf{G}'} V_{\text{L}}(\mathbf{G} - \mathbf{G}') c_{\mathbf{k}+\mathbf{G}',n} = \frac{1}{\Omega_{\text{c}}} \int e^{-i\mathbf{G}\cdot\mathbf{r}} V_{\text{L}}(\mathbf{r}) \sum_{\mathbf{G}'} e^{i\mathbf{G}'\cdot\mathbf{r}} c_{\mathbf{k}+\mathbf{G}',n} d^3\mathbf{r} \quad (1.49)$$

式 (1.20、1.21) より、 \mathbf{G}' についての足し合わせは

$$u_{\mathbf{k}n}(\mathbf{r}) = \sum_{\mathbf{G}'} e^{i\mathbf{G}'\cdot\mathbf{r}} c_{\mathbf{k}+\mathbf{G}',n} \quad (1.50)$$

と実空間の波動関数 (の格子周期部分) となる。従って、式 (1.49) の評価は、

1. 逆格子空間で展開された波動関数 $c_{\mathbf{k}+\mathbf{G},n}$ を実空間表現 $u_{\mathbf{k}n}(\mathbf{r})$ に変換する。
2. 実空間で $u_{\mathbf{k}n}(\mathbf{r})$ と $V_{\text{L}}(\mathbf{r})$ の積をとる。
3. その結果を逆格子空間上の表現に戻す。

こうして式 (1.49) の評価は、3 段階の演算になり、かえって大変になったようにも見えるが実は逆である。2 の演算は実空間の格子点での対角和であるからおよそ $\mathcal{O}(N_{\text{pw}})$ の演算で済み、1 と 3 は FFT を用いると $\mathcal{O}(N_{\text{pw}} \log(N_{\text{pw}}))$ の演算で済み、結果として全体でも $\mathcal{O}(N_{\text{pw}} \log(N_{\text{pw}}))$ のオーダーの演算で済むことになる。Car と Parrinello により擬ポテンシャル計算に対するこの技法が普及した [20]。

FFT を行う領域について述べておく必要がある。波動関数は、図 1.2 (a) のように半径 k_c の球内の N_{pw} 格子点で展開される。ところが電荷密度は波動関数の自乗で与えられるので、波動関数の変化を密度変化（そしてポテンシャル）に忠実に反映するにはその 2 倍の半径の逆格子点が必要である。同図 (b) 参照。その結果、FFT を行う矩形領域は一边の長さが $4k_c$ となる。逆格子空間における FFT メッシュと、実空間におけるそれとはだいたい同じであるから、結局式 (1.26) より、実空間の FFT メッシュ点の数 N_{FFT} は

$$N_{\text{FFT}} \cong 16N_{\text{pw}} \quad (1.51)$$

となる。

1.9 Hellmann-Faynman 力とストレス

Hellmann-Faynman の定理のおかげで原子間力とストレスを効果的に計算することができる。

結晶における、ある原子における力は、原子位置 \mathbf{R}_I に関する全エネルギーの微分で与えられる。

$$\mathbf{F}_I = -\frac{dE_{\text{tot}}}{d\mathbf{R}_I} = -\frac{d}{d\mathbf{R}_I} \langle \Psi | \hat{H} | \Psi \rangle \quad (1.52)$$

この定義によると、数値的に原子間力を求めるには、全エネルギーを原子位置に関して何点か（最低 2 点）で評価しなければならないことになる。

ところが、Hellmann-Faynman の定理によると、式 (1.52) は、

$$\mathbf{F}_I = -\langle \Psi | \frac{\partial V(\{\mathbf{R}\})}{\partial \mathbf{R}_I} | \Psi \rangle \quad (1.53)$$

と変形できる（引用としては当然 Hellmann と Faynman 両方をするべきであるが、作者は Faynman[21] のものしか持っていないので Hellmann さんごめんなさい）。一見、式 (1.52) と (1.53) はちょっとした違いにしか見えないかもしれないが、実は大変な違いがある。式 (1.52) の中で、波動関数 Ψ は原子位置に陰に依存する。原子位置が変わると、 Ψ は改めて評価し直さなければならない。それに対し、式 (1.53) の中では、波動関数 Ψ は平衡位置のものである。従って平衡位置での Schrödinger 方程式を一回解けば十分である。一般にはポテンシャルの位置微分を取り空間積分するほうが、別の位置における Schrödinger 方程式を改めて解くよりはるかに簡単である。

力とストレスは特に平面波を用いた擬ポテンシャル近似で効果的に評価できる。基底関数が空間に固定されているからである。その平面波展開の具体的表式は、力は [14] に、ストレスは [22, 23] に与えられている。（文献 [23] の式 (2) に現れている運動エネルギーの前の符号は - のタイプミスである。）

力の情報はエネルギーを最小にする単位格子中の原子位置決定に使われる。同じようにストレスは格子定数の最適化に使われる。

系の全エネルギーの計算よりも力やストレスのほうが収束が遅いことに注意すべきだ。全エネルギーはセルフコンシステント解に到達したとき最小値になり、その時波動関数の変位に対して変分量となる。すなわち、1 次の波動関数の変位に対して、全エネルギー変化は 2 次となる。この意味では力やストレスは変分量ではなく、波動関数の変位の 1 次として変化する。

第2章 ロードマップ

2.1 概略

擬ポテンシャル平面波法電子状態計算プログラム「Osaka2002」は多くのプログラムの集合体である。それらの間の関係は次のようになっている。

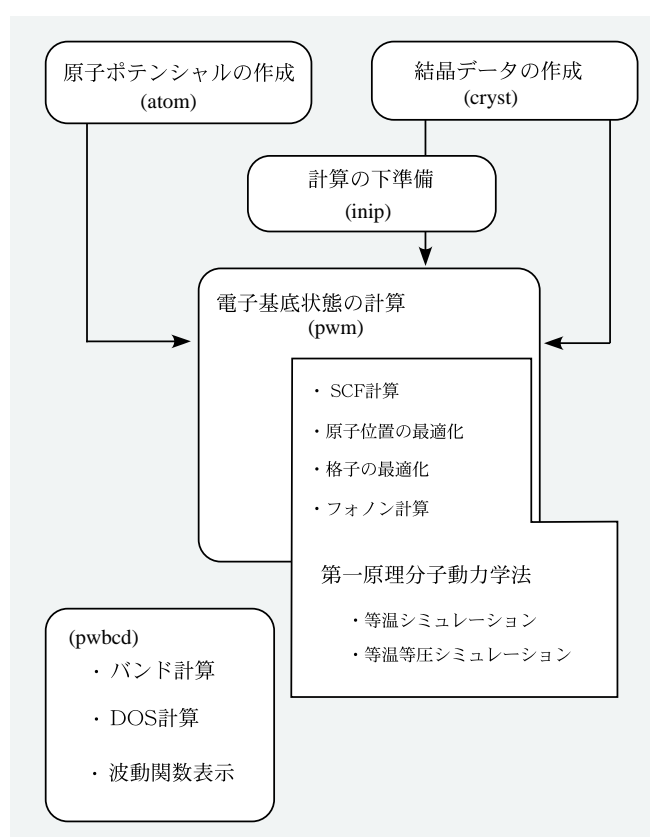


図 2.1: パッケージ Osaka2002 で行う計算

簡単に言って、atom で計算に必要な原子種のポテンシャルを作り、cryst で結晶を作り、pwm でセルフコンシステント (SCF) 計算を行う。inip は pwm の計算の下準備をする。

パッケージ「Osaka2002」の中核は pwm で、この中 SCF 計算が完成すればそれからさまざまな発展が出来る。それらの中に格子構造の最適化、フォノン計算、さらに第一原理分子動力学シミュレーションなどがあるが、それらは pwm の中に一部として実行される。旧バージョンの第一原理分子動力学シミュレーション mds はこの pwm の中にいくつかの

ジョブの一つとして含まれるようになった。

電子状態の詳細スペクトルが必要であればバンド計算、DOS 計算などを行うことになるが、それは pwbcd が担う。pwbcd は旧バージョンで別々となっていたバンド計算、DOS 計算、波動関数表示のプログラムを統合したものである。

atom は擬ポテンシャル生成プログラムでこれ自身、独立したものである。cryst は結晶を作るプログラムである。

全体を通じて、結晶の対称性を利用する部分は TSPACE[24] と呼ばれるライブラリーが支える。

2.2 計算環境と準備

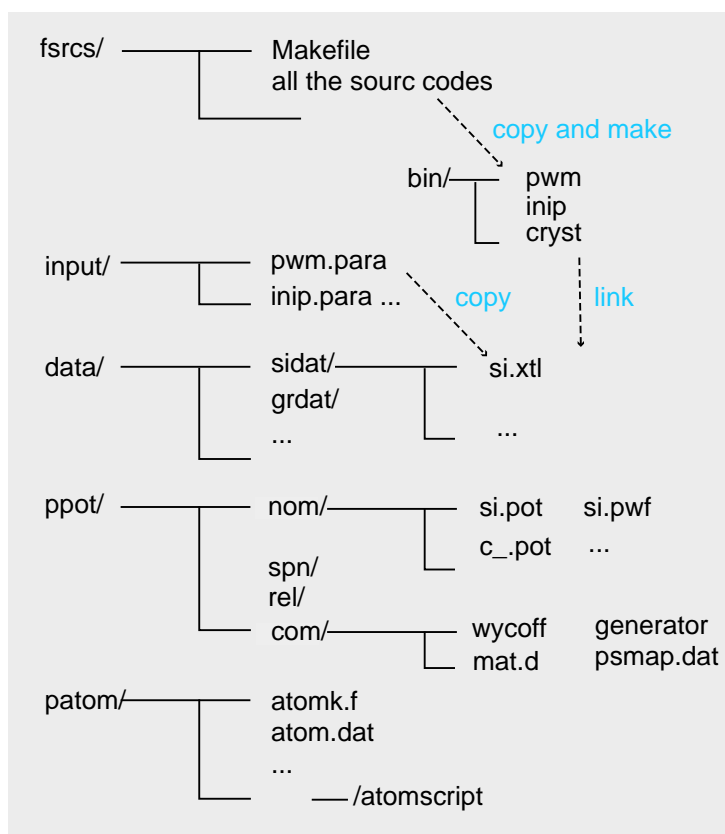


図 2.2: パッケージ Osaka2002 のディレクトリ構造

ファイル構成

ソースプログラムを持ってきた後、適当なディレクトリーで解凍する。ファイルは図 2.2 で示されるように、解凍した時点で 4 つのディレクトリーができ、さらにユーザーが /bin ディレクトリーを作成しここに実行可能プログラムを格納する。この図に載っていないがその他 /drivers というディレクトリーがある。ここに機種依存を持った特別のファイルや、機種ごとの Makefile が収められている。

全てのソースコードは Makefile とともに、ディレクトリー/fsrsc に置かれている。それを/bin ディレクトリーにコピーしこのディレクトリーでコンパイルし実行可能ファイルを作成する。実行可能ファイルができれば、あとはそれらを計算したいデータディレクトリー、例えば sidat/ にリンク、コピーして使うことになる。

ディレクトリー/input にはプログラムの入力ファイルのテンプレートが置かれている
ユーザーから見て Osaka2002 の前バージョンと一番大きな違いは、pwm を初めとしてほとんどのプログラムが市販のアプリケーションプログラムと同じように一度コンパイルするとそれで十分で、結晶ごとに再コンパイルの必要がないことである。以前はプログラムに使われている配列のディメンジョンを include ファイルで指定する必要があったので結晶ごとに作り直さねばならなかった。今回では作られたバイナリープログラムは/bin にまとめておいて、どのような結晶でも共通して使うことができる。

コンパイル

まず/drivers から自分の機種に合った Makefile、drivs.f90 (必要に応じてさらに fft ルーチンなど) を取り出しソースディレクトリー fsrsc に入れる。そして drivs.f90 に対して後で述べるような編集をする。更なる機種ごとの対応については「Osaka2002 のインストールマニュアル」を参照されたい。

Makefile ファイルを/bin ディレクトリーにコピーし

```
% make prep
```

とコマンドを打つ。これにより必要な全てのソースコードがディレクトリー bin にコピーないしはリンクされる。

コンパイルが終わったら、

```
% make clean
```

ないしは

```
% make clobber
```

として、不必要なファイルを消しておく。

Makefile に必要な情報が書き込まれておればコンパイルはコマンド一つで済む。inip であれば

```
% make inip
```

pwm であれば

```
% make
```

となる。

バンド計算、DOS 計算、波動関数計算のプログラム pwbed のコンパイルは


```
% make pwbcd
```

である。

原子ポテンシャルは、ディレクトリ `patom/` で作成し、作られたデータファイルは `ppot/` のディレクトリにその種類ごとに格納しておく。

2.2.1 機種ごとの対応

ディレクトリに必要なファイルを準備したら、機種ごとの設定を行う。ディレクトリ `fsrcs/` の `drivers/` に機種ごとに作ってある `Makefile` や `drivs.f90` が入っている。適当なものを `fsrcs/` のレベルに取りだし、編集する。

自分の機種に合った `drivs.f90` を取り出し、サブルーチン `potdir` を見つけ、その中の

```
CHARACTER(LEN=15) :: datadir = '/home/user/ppot'
```

の部分で自分の環境のポテンシャルデータディレクトリに変える。

`lapack` などの数値計算ライブラリーは当然、使用するシステムごとに違うので、そこは各自ユーザーが調べて `Makefile` 中にリンク方法を指示する。その他、`TSPACE` はコンパイラによって対処方法が異なるので「Osaka2002 のインストールマニュアル」の付録に述べる。また高速フーリエ変換の部分のサブルーチンは標準のものがついてくるが、様々なライブラリーのものを利用したければやはり「Osaka2002 のインストールマニュアル」の付録を参照のこと。

TSPACE の既存ユーザーへ

本パッケージに添付している `TSPACE` は文献 [24] に付属しているものと同じだが、ただし取り扱う原子数などのパラメータを柔軟に変えられるようにするため、いくつかのパラメータは別ファイル `TSPARAM` として用意し (4.3 節参照)、それを `TSPACE.F` が読み込むようにしている。そのため本パッケージに添付しているものを使うべきである。

以上でファイル構成の準備は終了で、後は結晶を作る原子全てのポテンシャルを作りディレクトリ `ppot/` に格納する必要がある。次章ではその作り方について説明する。

第3章 原子擬ポテンシャル

擬ポテンシャル法の特徴はポテンシャルを作る部分と、それを使う部分とが独立していることである。現在擬ポテンシャルを作るプログラムはいくつか流布されているが、Osaka2k パッケージは擬ポテンシャルについてはそのようなポテンシャル作成プログラムを利用させてもらって、それをもっぱら利用する立場に立っている。したがって Osaka2k パッケージにあって、擬ポテンシャルを作成する部分のプログラム `atom` は著者らが直接責任を負えない唯一の部分である。

もっともほとんどのユーザーにとってその内部原理は問題でなく、使えればそれで十分であろう。その意味では Barkley 版をもとに Troullier、Martins が開発したこの `atom` は非常に完成度の高いプログラムで、ほとんどバグがとれていると言われているだけあって、著者らもこのプログラムにはかなりの信頼を置いている。Osaka2k の開発途中で、問題を `atom` まで遡って調べる必要の起きることはほとんどなかった。

このプログラムの用途は非常に広いが、ここでは `pwm` に用いることに限って解説する。ソースコードは `patom` ディレクトリーにある `atomk.f` である¹。 `atomk.f` を

```
f77 atomk.f -o atom
```

とコンパイルした後、扱いたい原子に対し一つ一つ擬ポテンシャルを作る²。

3.1 入力ファイル

まず入力ファイル `atom.dat` を編集するが、そのフォーマットは以下のようになる。

Table 3.1: `atom.dat` のフォーマット

<i>itype</i>	<i>ititle</i>				
	<i>ikerk</i>				
<i>nameat</i>	<i>icorr + ispp</i>				
<i>znuc</i>	<i>zsh</i>	<i>rsh</i>	<i>rmax</i>	<i>aa</i>	<i>bb</i>
<i>n_{core}</i>	<i>n_{val}</i>				
<i>n</i>	<i>l</i>	<i>zo_↓</i>	<i>zo_↑</i>	<i>evd</i>	
	...				
<i>r_{cs}</i>	<i>r_{cp}</i>	<i>r_{cd}</i>	<i>r_{cf}</i>	<i>cfac</i>	<i>rcfac</i>

¹オリジナルのコードは `atom.f` だが、`atomk.f` はその出力の部分だけを変えたものである。

²完成度が高いとはいえやはり機種依存がある。著者らが経験した範囲内での対処法はインストレーションマニュアルに書かれている。

それぞれの行の意味を書くと、

一行目 計算種 (*itype*) とタイトル (*ititle*)。計算種としては

itype=

- **ae:** 全電子計算
- **pg:** 擬ポテンシャル生成
- **pe:** 交換項に関する内殻補正をした擬ポテンシャル生成
- **ph:** ハートリー項まで含めた内殻補正をした擬ポテンシャル生成
- **pt:** 擬ポテンシャルテスト
- **pm:** 擬ポテンシャルのテストと価電子の修正

二行目 擬ポテンシャルの種類 (*ikerk*)

ikerk=

- tm2 Improved Troullier and Martins
- bhs Bachelet, Mamann, Schuter
- oth generate data file
- van Vanderbilt
- tbk Troullier and Martins
- yes Kerker
- no Hamann, Schluter, Chiang

三行目 原子名 (*nameat*) と計算オプション (*icorr + ispp*)。 *nameat* は原子記号で与える。 *icorr + ispp* は連続した3文字である。初めの2つは *icorr* で、電子相関関数の種類を与える。選択肢としては

icorr=

- ca Ceperly-Alder (Perdew-Zunger parameterization)
- xa $X\alpha$ method
- wi Wigner interpolation scheme
- hl Hedin-Lundqvist
- gl Gunnarson-Lundqvist-Wilkins
- bh von Barth-Hedin

がある。

3文字目の *ispp* は、

ispp=

- r relativistic calculation
- s spin polarization

□ none

を選択する。

四行目 核の電荷数 ($znuc$)、内殻の電荷数 (zsh)、内殻半径 (rsh)、最大半径 ($rmax$)、
動径方向の刻みのパラメータ (aa 、 bb)、

五行目 内殻軌道の数 (n_{core}) と価電子軌道の数 (n_{val})、

六行目以下 n と l で指定される価電子軌道ごとの電子占有数 (zo)、オプションとして
軌道エネルギー (evd)、

最終行 擬ポテンシャル生成のカットオフ半径 (r_c)、およびオプションとして、内殻補
正パラメータ ($cfac$ と $refac$)、

原子軌道はスピンを考慮しなければ、本来は主量子数 n と角運動量子数 l と共に、方位量子数 m も与える必要があると疑問に思われるかもしれない。開殻原子に対しては本来その通りだが、このプログラム (他のものもほとんどそう) では、ポテンシャルは電子の寄与も含めて完全な球対称を仮定している。すなわち、三つある p 軌道、 p_x 、 p_y 、 p_z に電子が一個しかない場合でも、電子は例えば p_x だけを占めていると考えのではなく、 p_x 、 p_y 、 p_z それぞれを平等に $1/3$ ずつ占めると仮定している。でなければ問題は本質的に3次元問題となり解くことが急速に難しくなる。従って m は考慮しない。

以下 Si 原子を例にとりその書き方を説明する。

```
pg      Silicon
      tm2
n=Si c=ca
      0.0      0.0      0.0      0.0      0.0      0.0
      3      2
      3      0      2.00      0.00
      3      1      2.00      0.00
      2.13      2.57      1.50      1.50
```

第一行目は、計算の種類コードと、原子の名前、第二行目には、擬ポテンシャルの種類が来る。ここでは

pg 擬ポテンシャル計算

tm2 Improved Troullier and Martins

を選択しておく。

第四行目は無視。このままにする。

五行目は、主量子数 n 、角運動量子数 l で区別された内殻軌道の数、価電子軌道の数
が来る。Si の例では、中性原子の基底状態の電子占有は $(1s)^2(2s)^2(2p)^6(3s)^2(3p)^2$ のよう
になっている。つまり内殻軌道は $1s$ 、 $2s$ 、 $2p$ の3つ、価電子軌道は、 $3s$ 、 $3p$ の2つなの
で例のような入力となる。

次に価電子軌道の状態が、一行に、 n 、 l 、スピンドウン、アップの占有数の順で記述される。今はスピンを考えないので、スピンドウンのみの占有数を使う。今の Si の例では、 $3s$ に 2 個、 $3p$ に 2 個の電子を占有させている。

最後の行が擬ポテンシャルをつくる時のカットオフ半径の値を s 、 p 、 d 、 f 軌道の順で並べる。単位は原子単位である。

3.2 実行

ここでは最も基本的な使い方を説明する。さらに進んだ取り扱いについては下巻参照。

1 全電子計算 入力データで計算の種類コードは擬ポテンシャル生成 pg とし³、atomk を実行。

また最後の行のカットオフ半径のデータは初めは適当な値にしておく。atom.dat ファイルを atomk と同じディレクトリに入れ、atomk を実行。

2 出力ファイル atom.out を読む 全電子計算の結果が出てくる。各軌道エネルギー値とともに、動径方向軌道の形に関する情報も書かれている。この時点で実際に軌道の図を見ることを勧める。それにはファイル fort.11 を適当なプロットングソフトで読めばよい。

ファイル atom.out では

```
radial grid parameters
```

の後、各軌道の動径方向軌道の特徴が数値データで示されている。例として、

```
n = 2  l = 0  s = .0
a extr      .634  -1.373
r extr      .057   .458
r zero      .153
r 90/99 %   .903   1.348
```

を見る。これは $2s$ 軌道に関するものである。この軌道は、2つの極値をもち、原点以外に零点が1つある。零点の位置が r zero で与えられる。r extr は2つの極値点の位置（原子単位）、a extr はその極値である。ユーザーはこのように各軌道関数が期待どおりの振る舞いをしているかチェックすべきである。

次に、価電子軌道の形に注目してカットオフ半径を決める。カットオフ半径 r_c の選択には任意性があるが、だいたいの目安として、最外殻の極値点の 1.1 ~ 1.6 倍くらいをとれば良い。波動関数のノード点に近くなければその値が 1 より小さくても構わない。一般に、 r_c が大きくなればなるほど、ポテンシャルは緩やかになり収束は早くなるが、その代わりに transferability は悪くなる。Troullier-Martins タイプの擬ポテンシャルはこの r_c の値が他のタイプのもの（例えば Bachelet らのもの [12]）に比べて大きく取れることが特徴である。このようにして決められたカットオフ半径を改めて atom.dat に書き込む。

³全電子計算は本来 ae の選択をすべきだが、pg でも計算の前段で全電子計算をするので結果は同じ。

3 擬ポテンシャルの生成 出力ファイル atom.out など全て消してから再度 atom を実行。これで擬ポテンシャル、および付随する擬波動関数が生成される (ファイル fort.10)。この時点で両方ともグラフ化して眺めてみるのが望ましい。

fort.10 のフォーマットは波動関数、ポテンシャルの順に、そしてその各々の中で、軌道として s 、 p の順で、各データが r 座標、波動関数/ポテンシャルの値の 2 行にわたりリストされる。冒頭には r のセグメントの数が書かれている。出力される擬ポテンシャルは電子密度の寄与を除いた裸のイオンポテンシャルで、この動径波動関数 $u(r)$ はそのまま規格化されている。すなわち

$$\int_0^{\infty} |u(r)|^2 dr = 1 \quad (3.1)$$

であり、 $\int \dots r dr$ となっていないことに注意。

4 ファイルの名付け 最後に、こうしてできたデータファイルはたくさんあるが、そのうちポテンシャルを記述する pseudo.dat01 および擬波動関数を記述する fort.13 を以降の計算で用いる。それぞれ原子の名前に変えてディレクトリ ppot/nom に格納する。名前はポテンシャルが (元素記号).pot で、波動関数が (元素記号).pwf である。今の場合 si.pot および si.pwf となる。全て小文字にすること。C のように一文字の場合は、c_.pot のように_をはさむ。

5 くり返し 以上のステップを欲しい原子だけ繰り返し替えず。その際出力ファイルは atom が動作するディレクトリーから取り除いておくこと。でないとエラーを起こす。

価電子の取り方

ポテンシャル生成の時、どこまで価電子とみなすかという問題がある。Si のように、 s 、 p 軌道のみでなる原子では単純であった。原子基底状態そのままの軌道を取ることで良かった。

しかしその場合でも Na だと価電子が s 軌道一つで、そうすると生成される擬ポテンシャルは s 軌道に対するもの一つとなってしまう。これでは上の軌道に対する精度が悪いので、この場合は価電子占有数として、 s 軌道に 0.75、 p 軌道に 0.25 というふうに入力し、 s 、 p 軌道両方に対する擬ポテンシャルを生成するほうが望ましい。

遷移金属など d 、 f 電子が入った場合の取り扱いは、ポテンシャルの生成で特別な注意が必要である。それに関しては以降の巻で取り扱うことになる。

第4章 結晶の作成

4.1 結晶データ入力

結晶はその単位格子、およびその中の原子位置によって記述される。単位格子は、その中で最小のものは基本単位格子 (primitive unit cell) と呼ばれるが、実際にはいろいろな理由で慣用単位格子 (conventional unit cell) が多く用いられる。primitive unit cell は通常その単位並進ベクトルを \mathbf{R}_1 、 \mathbf{R}_2 、 \mathbf{R}_3 と表すのに対し、conventional unit cell はその単位並進ベクトルを \mathbf{a} 、 \mathbf{b} 、 \mathbf{c} と表す。座標 (h, k, l) というとき、何も書かれていないと、それが $h\mathbf{R}_1 + k\mathbf{R}_2 + l\mathbf{R}_3$ を意味するのか $h\mathbf{a} + k\mathbf{b} + l\mathbf{c}$ を意味するのかわからない。そこで、ここでは前者を primitive base 表現、後者を conventional base 表現と呼び区別することにする。それにともない、逆格子も、primitive base では \mathbf{G}_1 、 \mathbf{G}_2 、 \mathbf{G}_3 、conventional base では \mathbf{a}^* 、 \mathbf{b}^* 、 \mathbf{c}^* と表す。

一般的な結晶データは、この意味で conventional base で表され、3つの基本ベクトルの長さ a 、 b 、 c 、それらの間の角 α 、 β 、 γ で表される (α は b 、 c 軸の間の角。以下同様)。cryst では、 a 、 b 、 c 、 α 、 β 、 γ を用いて慣用単位格子のベクトルを直交座標系で、

$$\begin{aligned} \mathbf{a} &= a\hat{x} \\ \mathbf{b} &= b(\cos\gamma\hat{x} + \sin\gamma\hat{y}) \\ \mathbf{c} &= c(\cos\beta\hat{x} + r\hat{y} + \sqrt{1 - \cos^2\beta - r^2}\hat{z}) \end{aligned} \quad (4.1)$$

ここで

$$r = (\cos\alpha - \cos\beta\cos\gamma) / \sin\gamma$$

と取っている。ただし菱面体結晶に関しては、式 (4.1) のようにとったのでは、3回軸が z 方向とならない。それでこの場合は例外として、3回軸を z 方向に持つように座標系を取っている。その取り方を図示しておく (図 4.1)。2回回転軸は x 方向に取っている。

基本的には計算結果はこの特定の座標系の取り方に依存しない。しかしフォノンの固有ベクトルや弾性定数など、ともかくも座標系を指定しないことには記述不可能なものもある。これらに関する座標系は今述べた取り方をしている。

ところで TSPACE は基本的には conventional base で書かれているが、pwm はもともと TSPACE とは独立して開発されたため primitive base の記述が基本となっている。このため Osaka2k のソースコードにはいたる所、座標系変換のコードが入れられていて非常に読みにくくなっている。

結晶データは、name.xtl というファイルに書かれている。例として挙げると、GaAs を記述したファイル gaas.xtl は、

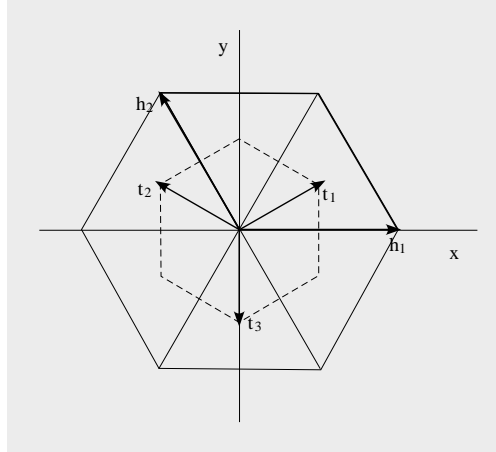


図 4.1: 菱面体結晶の軸の取り方。2 回軸は x 軸、鏡映面は y 軸にとっている。

```

TITLE GAAS
DIMENSION 3
CELL
    5.65315    5.65315    5.65315    90.00000    90.00000    90.00000
SYMMETRY NUMBER 216 QUALIFIER ORIGIN_1

```

```

ATOMS
NAME      X          Y          Z      POT  CHARGE  TEMP  OCCUP  SCAT
GA1      0.00000    0.00000    0.00000  ga   3.0000  0.5000  1.0000  GA
AS1      0.25000    0.25000    0.25000  as   5.0000  0.5000  1.0000  AS

```

となる。このフォーマットは MSI 社の INSIGHTII の入力フォーマットに従ったものである。従ってこのデータは INSIGHTII でそのまま表示できる。

ファイルフォーマットは、TITLE、DIMENSION 3 の後、格子定数パラメータが続く。標準的な格子定数の記述の仕方に従って、

$$a, b, c, \alpha, \beta, \gamma$$

の値が、Å あるいは deg の単位で並べらる。標準的という意味は、conventional 表現を用いるということである。

続いて、空間群の番号が書かれ、必要に応じて後で述べる QUALIFIER をつけることができる。

その後各原子の座標が、相対座標 (conventional unit cell に対して) で書かれる。大抵の結晶データは conventional base での相対座標で表されているので、この処置は適切であると信じているが、三方晶の場合はしばしば混乱があるので注意が必要だ。三方晶の場合の問題は文献 [26] で整理されているので参考になると思う。ブラベー格子が菱面体なのは QUALIFIER を Rhombohedral とすることが基本であるが、その座標を六方晶として取っているものもあるので、その場合は QUALIFIER を Hexagonal とする。

ここで単位格子内の原子数に関するいくつかの量を区別する必要がある。¹

N_{at} : 全原子数

N_{spe} : 原子種類数

N_{ka} : 既約サイト数 すなわちその結晶の対称性では結ばれない原子の数

N_{spe} と N_{ka} の違いは、元素として同じものは前者では一つとして数えるのに対し、後者では対称性で結ばなければ別種として数えるということである。例としてグラフィトを考えると

$$N_{\text{spe}} = 1, N_{\text{ka}} = 2, N_{\text{at}} = 4$$

となる。

ファイル*.xtl でリストされているものは、単位格子内の全ての原子ではなく、既約サイト N_{ka} の原子に対するものだけで良い。残りの原子はこれらの既約サイトから空間群の対称操作により作られる。これらの量の識別は、付属するサンプル結晶データファイルで練習すると良い。

NAME は原子の名前とは必ずしも一致する必要はなく適当につけてかまわない。次に相対座標が書かれ、その後使うポテンシャルデータの名前が来る。この名前が実質使う原子を指定している。この名前により前章でつくったポテンシャルのファイル名を探すので、小文字であるべきである。つまり Si に対しては si とし、C に対しては c□ と空白を入れる。

それ以降のデータはここでは読まれないので無視する。²

空間群の QUALIFIER としては、"International Tables for Crystallography" (ITC) [25] の記述に従う。出てくるものは、

ORIGIN_1 か ORIGIN_2

三方晶に関しては

RHOMBOHEDRAL か HEXAGONAL

対称性が低い結晶では、

UNIQUE_b か UNIQUE_c

の選択の他、それに加えて

UNIQUE_b, CELL_1 or 2 or 3

がある。ただしこのパッケージの空間群を受け持つ TSPACE の制限より、底心中心の場合は C center しか受け付けないから、そうっていないものは自分で焼き直す必要がある。そして ITC をみて C center となっている CELL_n を選ぶべきだ。

¹基本的にはインクルーズファイルが無くなった今では以下のパラメータを陽に指定することは無くなったが、依然そのパラメータの意味を知ることが重要なので、以下にその記述を残した。実際のところ本体プログラムからはインクルーズファイルは無くなったが、TSPACE は依然インクルーズファイルを使っている。

²以前のバージョンではこの後の価電子数も指定する必要があったが今はその必要がなくなった。それらの値はポテンシャルデータから読まれる。

4.2 cryst の実行

次にこの*.xtl を用いて cryst を実行する。その結果*.prim が出力される。以降の全ての計算において、結晶データは*.xtl を参照するのではなく、この*.prim を参照する。またこれ以降の計算内部では、単位は基本的には原子単位になることも記憶しておいたほうが良い。出力時のみ必要に応じて他の単位系も用いられるが、何も表示のないものは原子単位である。

使い方は、今の例では、si.xtl をディレクトリ sidat/ に置き、cryst を実行する。結晶名を聞かれるので、

```
input the crystal name with a period at the end.
> si.
```

と入力する。ピリオドを忘れずに。

出力されたファイル si.prim を必ずチェックすること。タイトル、日付データの後、conventional unit cell、primitive unit cell、逆格子ベクトルのそれぞれの3軸が直交座標で表示される。

```
TITLE GAAS
      date:      Fri Jan  5 17:56:20 2001
DIMENSION 3
LATTICE PARAMETERS (A,B,C,CA,CB,CC) in a.u.
      10.6829045  10.6829045  10.6829045
      0.0000000  0.0000000  0.0000000
Space group
      216  Td2  F-43m  ORIGIN_1
IL  NG  NC
  2  24  1  ORI
The conventional vectors
      10.6829045  0.0000000  0.0000000
      0.0000000  10.6829045  0.0000000
      0.0000000  0.0000000  10.6829045
The primitive vectors
      0.0000000  5.3414522  5.3414522
      5.3414522  0.0000000  5.3414522
      5.3414522  5.3414522  0.0000000
The primitive reciprocal vectors without 2Pi
      -0.0936075  0.0936075  0.0936075
      0.0936075 -0.0936075  0.0936075
      0.0936075  0.0936075 -0.0936075
```

IL、NG、NC は TSPACE のパラメータで、それぞれ格子のタイプ、結晶点群のオーダー、座標原点の選択数である。

入力された空間群の番号に基づいて、primitive unit cell の情報が計算されるが、それが予想通りになっているかチェックすべきである。次に N_{spe} とその原子情報が現れる。そのあと、原子の既約サイト数 N_{ka} が来て、それぞれについてのワイコフ位置の記述が続く。

```
Number of atom species
  2
```

```

No      Name      Zat      Zval
1       ga       31       3
2       as       33       5
KIND OF ATOMS
2
Wyckoff Positions
ATM (    x,      y,      z)      Nos      Wycf      Code
1 ( 0.00000, 0.00000, 0.00000)  1/ 1      4a      0 0/1  0 0/1  0 0/1
2 ( 0.25000, 0.25000, 0.25000)  3/ 1      4c      0 1/4  0 1/4  0 1/4
NUMBER OF ATOMS
2
L.L. AND U.U.      VALENCE      ELEMENT
1      1      3.0000      1      ga
2      2      5.0000      2      as
POSITIONS RELATIVE TO A UNIT CONVENTIONAL CELL      SPECIES      SYM(IG)
1      0.0000000      0.0000000      0.0000000      1      ga      1
2      0.2500000      0.2500000      0.2500000      2      as      1

```

その後、総原子数 N_{at} に続き、L.L. AND U.U. の見出しの後、既約サイト毎に、それに属している原子の区切り番号をリストしている。番号 L.L. から U.U. までの原子がその行の原子種類であることを示している。その行に表れる価電子数をチェックすべきだ。そして各原子位置の相対座標が現れる。

注意 `cyrst` は入力データからワイコフ位置を解釈するが、230 もの空間群の複雑さのため今のプログラムでは完全には解釈できない。³ 解釈に失敗すると `cryst` は常にその位置に一般点を当てる。従ってこの部分はユーザーが常に注意しなければならない部分である。間違ったワイコフ記号が割り振られていたら、手でその部分を直さなければならない。ワイコフ記号は ITC に従う。その位置コードは、TSPACE[24] の記号に従っている。すなわちそれは 3 つの組で表され、それぞれの組では、整数 + 分数が来ている。整数部分は 0 は相対座標が 0、1 は相対座標が一般値 x 、2 は y 、3 は z 、マイナスが付けば負の値を意味し、その後の分数はそのまま相対座標を示す。例えば、 $\{1\ 0/1\ 2\ 1/4\ -2\ 1/4\}$ とあれば、 $(x, y + \frac{1}{4}, -y + \frac{1}{4})$ を意味する。

4.3 制御パラメータ

これまでのところ、ユーザーが変更すべきパラメータはほとんどなかった。たいいていの目的には、TSPARAM と `xtl.inc` は初期値のまま問題ないと思う。しかし非常に大きなサイズの結晶などを使う場合にはこれでは間に合わず、ユーザーが変更しなければならないことがあるのでここで説明しておく。

TSPARAM は直接的には TSPACE が使うが、それ以降の `cryst`、`inip`、`pwm` などのほとんどの `include` ファイルがその設定値を伝承するので重要である。

```
PARAMETER (LMNATM=50,LMNKAT=20)
```

³この解釈がどれくらい大変なことか考えても見てくれ。例えば、相対座標で 0.5 の点がワイコフ座標で特種点 $1/2$ になるのか、それとも一般点 x でたまたまその値が 0.5 であったのか、それを 230 もの空間群の中で判定するのはどれくらい大変なことか。

PARAMETER (MAXNPW=4854)

LMNATM はセル内の全原子数 N_{at} の上限、LMNKAT は同じく既約サイト数 N_{ka} の上限である。従って、実際に使う数よりも大目にとっておく。

MAXNPW は平面波の数の上限であるが、これは TSPACE の中で使われる平面波の数で、後の pwm で使われる平面波数と別の量である。

4.4 結晶のグラフィックス表示

ここまで来ると、ユーザは結晶データが本当に正しいかを数字でなく図で確かめたいと思うだろう。これまでの数値データからだけで頭に結晶図が描ける人は幾何学能力がずば抜けて優れている人のみで、大抵の人は絵を見なければ想像もできないだろう。結晶が間違っているとこの後の計算が全て無駄となるので、結晶のチェックは必ず行う。`*.prim` のデータは Mathematica のノートブック `CrsytAnal.nb` でグラフィックス表示することができる。ファイル名 `*.prim` を入れればだいたい自動的に結晶を描くようになっているが、実際の結晶の複雑さのためユーザがどうしても描く範囲などを調整する必要はある。

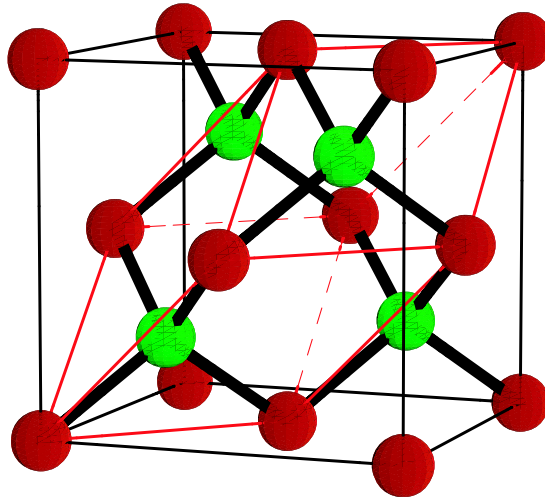


図 4.2: GaAs 結晶。赤線は基本単位格子を示す。

ここにその使い方の詳細を書くことは控えるが、だいたいノートブックのタイトルに沿いながら、ユーザが判断できると思う。またグラフィックス座標の取り方は付録 A を参照していただく。

`CrsytAnal.nb` は結晶を描くばかりでなく、ボンド間の距離、角度の計算などの解析に有用である。図 4.2 に `CrsytAnal.nb` による結晶図の例を示す。

第5章 電子の基底状態計算 (I)

5.1 inip

電子状態の計算プログラム `pwm` に入る前に、計算の前準備をいくつかしておく必要がある。それが `inip` である。¹

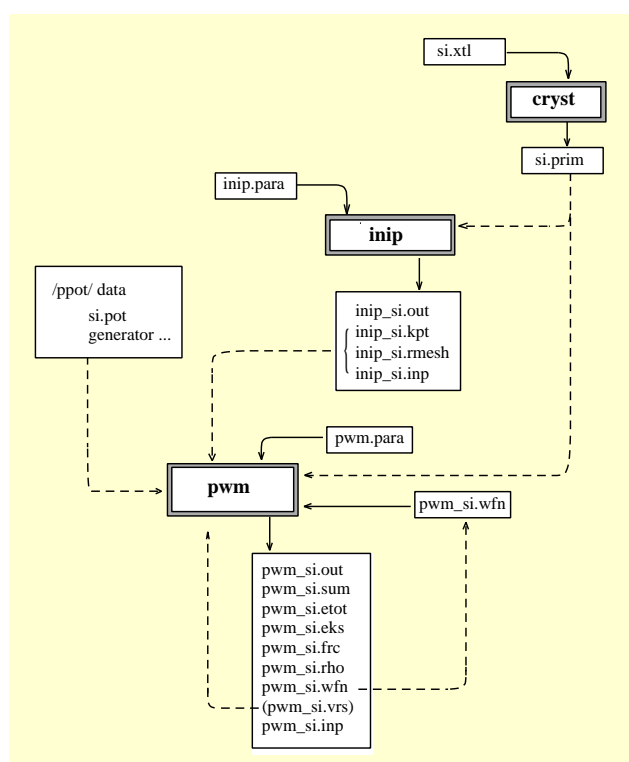


図 5.1: `pwm` を中心とするファイル依存性

`inip` の具体的な目的は、一つに逆格子空間でカットオフ半径 k_c を与え、その球内で平面波基底を展開すること。平面波数 N_{pw} を数え、それから適切な FFT 領域を割り出す。 k_c と FFT 領域を指定する `NGDIM` や `NG3` の間の関係は図 1.2 のようになる。FFT 領域の一辺が $2*NGDIM+1$ で、実空間では $NADIM=2*NGDIM$ となる。

`inip` のもう一つの目的は、 k サンプリング点を決定することである。これについては次

¹旧バージョンでは `inip` の一番大きな目的はあらかじめプログラムサイズを見積もり、`include` ファイル中のパラメータ指定をすることであった。新しいバージョンでは基本的に `include` ファイルを廃止しているので `inip` は必要ではなくなるが、しかしながら `inip` の役割はそれだけでなく、`pwm` での計算に必要ないろいろな下準備をする。これらは以前と同じく本体 `pwm` と切り離して行うことにしている。

節で述べる。

入力ファイル `inip.para` は以下ようになる。

```
Input file name (priod is needed at the end)
si.
Parameters about k points
Cutoff k radius (AMAX) given by lattice index without 2Pi
3.1
way to give sampling points (0:given manually, 1:calc)
1
number of k-sampling points
2
potential type (spin, NLCC, relativistic)
0      0      0
```

以下にそれぞれのパラメータの意味を順に説明する。

まず計算しようとする結晶の名前がピリオドとともに書かれる。その後、

- 平面波カットオフ半径 (AMAX)
平面波カットオフ半径 k_c を、逆格子ベクトルの中で最小の大きさ g_{\min} を単位に入力。
- k サンプルング点の与え方
1 で自動的に計算する。0 は手入力。
- k サンプルング分割数 (NKDIV)
自動計算のときの分割数。
- ポテンシャルに対する補正
ポテンシャルに対するオプションで、スピン、内殻補正、相対論的效果のオプションを付けるかどうかを指示する。1 は付ける、0 は付けない。これは擬ポテンシャル作成の `atom` に対するオプションと同じで、当然両者で一致すべき。

となる。

5.1.1 k サンプルング法

k サンプルング点の決め方には以下の方法がある。

(i) 自動計算 (一様サンプルング = デフォルト) 自動計算の場合は、1.6 節で述べられたように Monkhorst-Park の方法によりサンプルング点が決められる [16]。デフォルトではゾーンの 3 方向とも、 M を正整数として、 $(-M+1)/2M$ から $(M-1)/2M$ まで M 分割する ($2\pi/a$ 単位)。

つまり $M = 1$ で Γ 点のみ、 $M = 2$ で $(1/4, 1/4, 1/4)$ とそれらの符号を変えた計 8 点となる。実際には結晶の対称性のためサンプルングに必要な数はこれよりずっと少なくて済む。

図 5.2 にこの取り方の特徴を初心者の良くやる取り方と比較して示している。初心者であれば、まず差し当たり区切れの良い中心、次に両端、そしてそれらの間を等分することを考えるだろう。Monkhorst-Pack の方法のミソは境界点をなるべく避けていることである。その方がより少ないサンプリング点でも精度が保てるのである。

M の値としては、 $M = 1$ の場合を除き、偶数になることを推奨する。

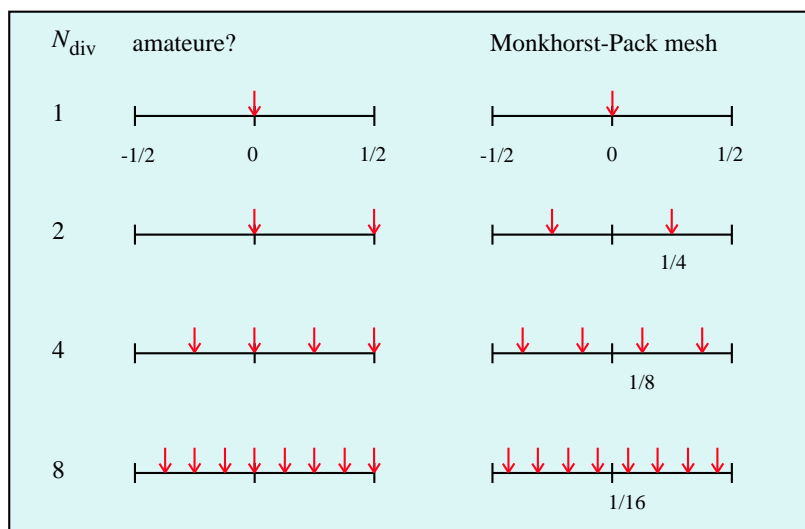


図 5.2: Monkhorst-Pack の方法によるサンプリング点の取り方と、初心者の取り方の比較。Monkhorst-Pack 法ではなるべくゾーン中心や、境界点が避けられるような取り方となっている。

この分割数 M の値を NKDIV に入力する。最終的なサンプリング点とその総数は、inip の出力ファイル inip_*.inp に書き出される。その部分は、

```
nkpts
  2
Sampling point in p
LD   -1  -1  -1/  4   WTK=  0.250000
XY    1  -1  -1/  4   WTK=  0.750000
```

のようになっている。つまり、 $M = 2$ に対して最終的なサンプリングの総数 n_{kpt} はたった 2 点で済んでいる。それらは、primitive base で $(-1/4, -1/4, -1/4)$ と $(1/4, -1/4, -1/4)$ で、それぞれの重みは $1/4$ と $3/4$ であることがわかる。

(ii) 自動計算 (非一様サンプリング) 超格子など異方性が著しい結晶では先の等方的サンプリングでは非効率となる。このようなときは異方性を考慮したサンプリングを取るべきである。そのような場合は、inip.para の基本データが入力し終わったところで 1 行開け、

```
OPTION BEGIN
```

```

anisotrop_ksample=
  1  1  4  4
OPTION END

```

のようにオプション指定をする。anisotrop_ksample=と与えることで、以前に指定した一様分割数 NKDIV の指定を無効にする。次の行で与えられる初めの3つの数字は基本ベクトル g_1 、 g_2 、 g_3 の方向の分割点数、最後の数字はその3つの分割点数の最小公倍数を与える。inip は与えた数字が最初の3つの最小公倍数であるかチェックしないので、これが間違っていると後の計算は誤ったものとなる。

(iii) 手動入力 自動計算は便利だが、一方こうした自動的な一様分割では扱えない場合がある。その場合はサンプリング点は自分で考え、手入力する。次がその手入力の例である。手動入力 k 点サンプリングの場合は上のリスト6～9行目のところを

```

way to give sampling points (0:given manually, 1:calc)
0
number of k-sampling points
4
KB(3), ICB
-1  -1  -1  4
  1   0   0  4
  0   1   0  4
  0   0   1  4

```

のように置き換える。これは primitive base で $(1/4, 1/4, 1/4)$ と $(1/4, 1/4, -1/4)$ の2点を与える。

六方晶などの場合は Chadi-Cohen アルゴリズム [30] に従ったサンプリングが有効な場合もあるので、このときは手入力にする。

k 点の数が少ない場合は、サンプリングの取り方は重要で、例えば同じ物質のさまざまな結晶多形を比較する場合、それらの多形間で「等価のサンプリング」をする必要がある。超格子構造や SiC がその良い例である [31]。

どちらの方法にしる、その k サンプリングの質がどれくらい良いものかを、式 (??b) で評価する必要がある。inip-*.out の

```

====> Final report for k-point sampling <====

```

のセクションで、

```

Quality of sampling by 2-points...
list only nonzero shells up to 41 shells
Sh No          shell sum
  8    ( -1.00000,  0.00000)
 15    (  1.00000,  0.00000)
 23    ( -1.00000,  0.00000)
 26    ( -1.00000,  0.00000)
 28    (  1.00000,  0.00000)
 32    ( -1.00000,  0.00000)
 36    (  1.00000,  0.00000)
 38    (  1.00000,  0.00000)

```

とリストされている。ここでは式 (??b) の値が0でないものだけを表示している。すなわちこのサンプリングでは第7近接の格子点までの和は0となるということだ。

5.1.2 平面波カットオフ半径

次に重要なパラメータは平面波カットオフ半径 (AMAX) である。この無次元 AMAX によりカットオフ半径 k_c は、

$$k_c = \text{AMAX} * g_{\min} \quad (5.1)$$

で、カットオフエネルギー E_{cut} (原子単位) は式 (1.25) で与えられる。 g_{\min} は3つの基本逆格子ベクトルの大きさの中で最小のものである。

それぞれのサンプリング点 \mathbf{k}_i について式 (1.24) を満たす全ての \mathbf{G} が平面波展開として取られる。従って平面波の数、 N_{pw} はサンプリング点ごとに少し違うことになる。具体的には、出力ファイル inip_*.out の中で、

```
=====> Final report for k-point sampling <=====  
For cutoff energy in Ry = 4.959185  
NHDIM = 59  
NGDIM = 4  
nkpts= 2  
Name: LD -1 -1 -1/ 4 NSTR= 8 WTK= 0.250000 NPW= 55 INV= 6  
Name: XY 1 -1 -1/ 4 NSTR= 24 WTK= 0.750000 NPW= 50 INV=18  
Sum over WTK 1.000000
```

の部分に書かれている。これから $(-1/4, -1/4, -1/4)$ と $(1/4, -1/4, -1/4)$ の2点では、 N_{pw} は55と50点と違った値となっていることがわかる。それぞれの点の全平面波のデータはファイル inip_*.kpt に収められる。inip はこのような情報などから、 N_{pw} の最大値としての NHDIM を59になるように指示している。この NHDIM とともに、pwm で使うパラメータの値は inip_*.out の最後の部分にまとめられている。

```
Passed parameters  
NEPC 8 NEDIM 4  
NKPTS 2 NHDIM 59  
NGDIM 4  
Then, ...  
LIN SEG NGLIN 9 NADIM 8  
VOL SEG NG3 729 NA3 512
```

このほか、inip は実空間の対称点に関するデータをファイル inip_*.rmesh に格納する。ただし結晶の対称性が全くない(すなわち空間群 $P1$) 場合は、このファイルは出力しない。

5.2 pwmの動作

「OSAKA2000」の核心部分 `pwm` は与えられた結晶に対して、セルフコンシステント計算を行い電子の基底状態をを求めるために用いる。理論の 1.7 節のところで述べられたように、それはエネルギー汎関数を直接繰り返し法により最小化することで達成される。

実行可能コード `pwm` を `make` で生成したあと、計算するディレクトリーにコピーあるいはリンクを取る。

5.2.1 セルフコンシステント計算

`pwm` の入力パラメータはファイル `pwm.para` で記述される。ここにユーザーがコントロールすべき基本的パラメータが書き込まれる。実際には `pwm` は他に多くの計算制御パラメータを持っているが、それらを全て `pwm.para` にリストすることは止めた。めったに使わないパラメータの使い方に余計な労力が割かれるからである。それらの付加的なパラメータは後で述べるオプションにより指定する方式を採用した。そうすることで上級ユーザーの高度な要求にも応えられるようなものにした。

入力ファイル `pwm.para` の典型的な例は以下のようなになるであろう。

```
Input file name
si.prim
Job (0:SCF Calc. 1:Atom optimize 2:Cell optimize 3:Phonon 4:MDS)
0
atom movement (0: OFF, 1: ON valid only when cell optimization)
0
number of iteration for electrons (maxIter0wfn)
15
conjugate-gradient paths (npath)
5
wftol
1.0D-11
ftol
8.0D-5
etol
1.0D-12
iread (previous WFs)
0
imoni,irhout,iwfout
0 1 1
Resume atom relaxation
0
max number of iterations for atoms (maxIter0atom)
4
Resume cell relaxation
0
max number of iterations for cell (maxIter0cell)
2

OPTION BEGIN
```

```
pressure_GPa=  
  0.0  
OPTION END
```

以下にそれぞれのパラメータの意味を順に説明する。

xtl.name	結晶名
Job	ジョブタイプ 0 : SCF 計算 (デフォルト) 1 : 原子位置の最適化、 2 : 格子の最適化、3 : フォノン計算、4 : MDS
atomMove0ctrl	原子位置の緩和 0 : しない (デフォルト) 1 : する
maxIter0wfn	波動関数に関するエネルギー汎関数の最小化過程の 繰り返し数の最大値
npath0wfn	一つのバンドに関する共役勾配繰り返し数
wftol	波動関数の収束までの許容値 (Ry^2 単位)
ftol	HF 力に関する収束許容値 (Ry/a_B^2 単位)
etol	エネルギー収束の許容値 (Ry 単位)
iread_wfn	初期波動関数の選択 0 : 入力なし (デフォルト)
imoni0ctrl	モニターオプション 0 : なし (デフォルト)
irhout	電荷密度ファイルの出力 1 : あり (デフォルト)
iwfout	波動関数ファイルの出力 1 : あり (デフォルト)
atomMove_resume	原子位置緩和の再開 0 : なし (デフォルト)
maxIter0atom	原子位置緩和の最大繰り返し数
cellMove_resume	結晶格子緩和の再開 0 : なし (デフォルト)
maxIter0cell	結晶格子緩和の最大繰り返し数

この中で最も重要なものは収束条件に関するパラメータである。理論のところ (1.7 節) で述べたように、電子の基底状態はエネルギー汎関数を繰り返し法によって最小化する。まず繰り返しの最小単位として、ある k_i 点において一つのバンドが取り上げられる。そのバンドにつき npath0wfn 回、共役勾配過程が繰り返される。それを全てのバンドについて一つ一つ繰り返す。そしてそれを全ての k_i 点で繰り返す。その全体の過程を maxIter0wfn 回だけ繰り返す。

この中でユーザーが最も頻繁に変えるものは maxIter0wfn であろう。npath0wfn は差し当たり 5 くらいの設定で良い。セルフコンシステント計算は繰り返しが maxIter0wfn を越えるか、波動関数の誤差が wftol 以下、あるいはステップごとのエネルギー変化が etol 以下になったときにストップする。

電子系のセルフコンシステント計算だけをもくろみたいときは、ジョブタイプは 0 とする。このときは atomMove_resume 以降のパラメータは無視される。

ある結晶に対して初めて計算するときには、iread_wfn=0 とする。このとき波動関数の初期値としては乱数を使われる。

出力制御として、モニタリング (imoni0ctrl)、電荷密度 (irhout)、波動関数 (iwfout) がある。デバッグ時以外は imoni0ctrl=0 とする。電荷密度、波動関数データファイルは必要なときは 1、不必要なときは 0 とする。大きな結晶では、これらのデータファイルが膨大になるので必要時以外は出力しないほうが良い。

オプション

オプションはユーザーの特別な要請に応じて追加するものである。これらの拡張機能はプログラムの改良とともに進化するので、個々の使い方はケーススタディとして付随するテクニカルレポートなどで述べるとして、以下に一般事項としての規則を述べる。

まず大文字 / 小文字の区別を守る。つまり case sensitive である。

一般条件が終わった後、1行を空け、

```
OPTION BEGIN
```

で始まり

```
OPTION END
```

で終わる。この間に余計な空白行は入れない。

オプションには2種類のタイプがある。

1. トグル変数

```
fermi_broadening ON
```

のように変数名の後、空白を一つ置き、ON/OFFをおく。

2. 値変数

```
pressure=
```

```
1.5
```

のように変数名の後、続けて=を置き、改行した後、少なくとも一つ以上の空白をおいた後、数値を入力する。1行目から値を入れてはいけない。実数タイプの数字では、たとえ300とぴったり整数値であっても、300.と表す。

現在このオプション指定で有効な変数のリストは(刻々変わるので)、実際のソースファイル auxinp.f90 を参照してもらおう。あるいはその入力部分をまとめた opts_list が参考になる。

これだけを準備できたら後は pwm の実行である。計算の短いものは、

```
% pwm
```

だけで良い。プログラムが立ち上がると計算経過がモニタに流れてくる。計算の長くなるものは、バックグラウンドで流すべきだ。

```
% pwm>stdout &
```

pwm は標準出力にいくつかの情報を打ち出すので、バックグラウンド時にはこのように出力のリダイレクトを行わないとエラーになる。大型などでキューシステムに乗せるには、そのマニュアルに従ってください。

問題とする結晶で、どれくらいの計算時間になるかを知ることは重要である。その目安を与えるものがディレクトリ becmrk/にある例である。これは8ヶの Si からなるダイヤモンド

ンドセルの計算を様々なマシンで実行したベンチマークが入っている。自分のマシンで比べてみてほしい。

5.2.2 解釈

pwm は計算が終わると図 5.1 のように様々なファイルを出力する。

pwm_*.out: 計算経過

pwm_*.eks: KS エネルギーレベル

pwm_*.frc: Hellman-Feynman 力

pwm_*.etot: 全エネルギー変化を中心とした計算の中核部

pwm_*.sum: 更なる要約

pwm_*.rho: 電荷密度分布

pwm_*.wfn: 波動関数。これはバイナリーファイル

このうち一番重要なのは pwm_*.etot である。その中身は、はじめに

```
===== CALCULATION PARAMETERS =====
```

JOB Type:

Self-consistent Electronic Calculation

Potential Type:

spin = 0 nlcc = 0 relativ = 0

KLEINMANN-BYLANDER TYPE FULLY SEPARABLE FORM

Ceperly-Alder type electron-correlation functional

maximum l = 1

...

のように計算条件の記述があり、その後結晶の記述が続く。それらに一応目を通しチェックすべきだ。

その後、

```
===== Selfconsistent Calculation =====
```

iter	Eel (Ry/cell)	deE (Ry/cell)	Xsi (Ry ² /cell)	nst/bk	aglmax
1	1.1167068599	-7.3979E+00	2.4698E-02	5/ 0	0.637356581
2	0.9919043641	-1.2480E-01	4.3798E-05	5/ 0	0.128972346
3	0.9916634496	-2.4091E-04	5.6158E-08	5/ 5	0.003105065
4	0.9916631217	-3.2783E-07	3.6162E-10	4/ 11	0.000173636
5	0.9916633948	2.7304E-07	2.3709E-11	2/ 6	0.000009096

CG process is stopped because increase in Eel 2.7304E-07

.....

Etot	Eel	delta E	resid	iter
=====	=====	=====	=====	===
-15.8051160963	0.9916633948	2.730E-07	2.371E-11	5

などと、電子系の収束の経過がリストされる。
各繰り返しごとに、

- iter: 繰り返し番号
- E_{el} : 電子系の全エネルギー。
- ΔE : 電子系の全エネルギーの前の繰り返しからの変化。
- ξ : 残留波動関数誤差 (式 (1.40) 参照)。
- パス数: 実際実行した共役勾配のパス数/内バックした回数。共役勾配のパス数は各バンドごと n_{path} 回繰り返されるが、繰り返しが進むと n_{path} 回繰り返す前に他の収束条件のためストップする。ここで現れる回数はバンドごとの値のうち最大値を示す。バックした回数というのは、共役勾配過程でエネルギーは基本的に減少するはずだが、時には失敗しエネルギーが増加することもある。そのようなときは、そこでそのバンドの共役勾配過程を止めて次のバンドに移る。バックした回数とは、全てのバンドに関するそのような事が起きた全回数である。
- θ : 波動関数更新時の新旧波動関数の混ざり具合を示す θ (式 (1.45) 参照)。各バンドの θ の内で最大値を示す。

が表示される。 E_{el} が減少し、それにつれ ΔE の絶対値、 ξ 、 θ が小さくなる様子がわかる。繰り返しは、 $|\Delta E| < etol$ となったため 5 回目でストップしている。このときの電子系のエネルギー E_{el} にイオンの Ewald エネルギー E_{ewald} を加えて、全結晶エネルギー E_{tot} が最後に出力される。

その次には、この全エネルギーの内訳が

```

component of Etot
kin          5.97858894
hart        1.08391474
exc         -4.78172988
loc         -1.80253465
nonloc      0.51342418
ewald       -16.79677949
-----
total       -15.80511616

```

のようにリストされる。これは全エネルギーを式 (1.22) のように分解したときのそれぞれの成分である。これによりどの項が全エネルギーに効いているかを知ることができそれなりに有用である。しかしながら、こうした項に分けることはある程度数学的な便宜上のことで、それにあまりに過大な物理的意味を持たせようとするのは危険である。

例えば、一番物理的意味がはっきりしていそうなイオン - イオン直接相互作用の Ewald 項 (式 (1.23)) でさえも、他の項との独立性がそう自明ではない。文献 [23] では、Si 結晶を圧縮したとき、全エネルギー変化を上のように分解し、Ewald 項は減少していることを示している。Si 結晶では、全ての原子が +4 価のイオンになっているわけだから、それらがお互いより密集したときクーロンエネルギーは上がると思うのが普通であろう。それ

は Ewald 項でさえも式 (1.22) のように分解したときの第 4 項の、イオンポテンシャルと電子の相互作用項と独立でないからである [27]。

最後にストレスが

total stress (Ry/Bohr³)

S(1,1)=	-2.24759E-05	S(2,3)=	3.17724E-10
S(2,2)=	-2.24759E-05	S(3,1)=	2.52820E-10
S(3,3)=	-2.24756E-05	S(1,2)=	-1.10795E-10

のように出力される。結晶の変形に対するストレスを求めて、弾性定数を求めることができる。ストレスに関しても、エネルギーのように各成分に分解することは可能で、それはファイル pwm_*.frc に書き出されている。しかしこの分解にあまりに多くの物理的解釈を持ち込むのは、エネルギーの場合と同様（もしかしたらそれ以上）危険である。

例えば、全エネルギーの式 (1.22) の第一項は運動エネルギーを表すが、それに「対応する」ストレスの表式は文献 [23] の式 (2) の中で、

$$\sum_{i, \mathbf{G}} |c_{\mathbf{k}_i + \mathbf{G}}|^2 (\mathbf{k}_i + \mathbf{G})_\alpha (\mathbf{k}_i + \mathbf{G})_\beta \quad (5.2)$$

となっている。この対応自身に何の問題もないが、しかしこの式 (5.2) が、文字通り、歪みに対する全エネルギーの式 (1.22) の運動エネルギー項の変化を与えるものとするのは誤りである。つまり両者は数値的に等しくないのである。Hellmann-Feynman の定理は、基底状態のセルフコンシステント解が求まったとき、「全」エネルギーの変化が、原子位置や格子定数などの「陽の」微分で与えられることを保証しているが、エネルギーの個々の成分については成り立たない。であるからストレスの表式も結局全体としてだけ意味を持つものである。

このほか、各原子に働く Hellmann-Feynman 力の値は、pwm_*.frc に出力される。

HELLMANN-FEYNMAN FORCE (RY/BOHR)

ATOM		X	Y	Z
ITER= 1				
	1 1	-0.230775E-08	-0.154249E-07	-0.573629E-07
	1 2	0.637597E-07	0.428339E-07	0.299777E-07

力の 3 成分は、直交座標を用いて表されているが、その座標系はもちろん結晶の並進ベクトルを取ったときの座標系である。

エネルギーの精度

ここで、電子エネルギー E_{el} の値は小数点以下 12 桁まで落ち着いている。そしてこの値は同じ計算条件で行えばどのマシンでもだいたい 8 桁以上までぴったり一致することに驚かれるのではないだろうか。しかし残念ながらこの桁がそのまま E_{el} の精度というわけではない。これはあくまで数値計算の上の安定性の問題で、 E_{el} の精度は別問題であるからだ。

E_{el} の精度が何で決まるかは難しい問題で、平面波カットオフエネルギーがこれこれしかじかの条件のとき E_{el} の誤差がかくかくしかじかになるという明確な判定基準があらかじめ予測できれば良いのだが、それを見つけないのが作者の悩みでもある。

数値計算における誤差の一般論から pwm の数値精度を評価してみよう。まず、 E_{el} の精度と言う場合、まず何を以て誤差というかがをはっきりさせなければならない。入力側には (i) 入

カデータの誤差、そして計算中では (ii) 計算機の数値精度、丸め誤算、(iii) 近似誤差、例えば積分における離散化誤差 (discretization error) が挙げられる。これらの誤差が計算中に増幅し、最終結果の誤差として現れる。これは誤差の伝播 (error propagation) と呼ばれているものであり、数値的安定性 (numerical stability) とともに数値計算上重要な概念である (詳しくは文献 [29] の第一章で説明されている)。

まず、入力誤差として、結晶構造データや、ポテンシャルデータの誤差が考えられるが、ここではそれらは一切考慮せず、入力した数値きっちりの精度、すなわち FORTRAN の倍精度で表されているとする。

(iii) の離散化誤差を評価してみる。全エネルギーの表式 (1.22) には様々な項があり、それらを全て調べる必要はあるが、簡単のため、イオン-電子間相互作用の部分だけで代表させよう。それは実空間積分 $\int \rho(\mathbf{r})V(\mathbf{r})d^3\mathbf{r}$ で与えられるが、積分はもちろん有限和として評価される。一般に微小区間 $\{-h, h\}$ での積分 $\int_{-h}^h f(x)dx$ を単に $2hf_0$ として与えたときの相対誤差は、

$$\frac{1}{6} \frac{f_0''}{f_0} h^2 \quad (5.3)$$

で与えられる。積分 $\int \rho(\mathbf{r})V(\mathbf{r})d^3\mathbf{r}$ で、非積分関数は、ポテンシャルと、電荷密度で与えられるが、ここでは簡単にポテンシャルだけと考える (ρ は実効的ポテンシャルをより滑らかにする働きをすると考えられる)。Si 結晶の場合、その積分はボンドの中心領域が支配するとして、そこでのポテンシャルの値とその2次微分の値を使うと、式 (5.3) の h^2 にかかる係数は ~ 0.05 と見積もられる。実空間でのメッシュ幅 h は格子定数を a_0 としたとき a_0/NADIM となる。Si では $a_0 \sim 7$ であるので、NADIM=16 の時、相対誤差は 1×10^{-2} と評価できる。NADIM=40 でようやく相対誤差は 1×10^{-3} と一桁上がる。

実際に、Si 結晶に対して、 E_{cut} を変えてみて、 E_{el} がどれくらい落ち着くかを見ている。表 5.1 にその数値実験結果が出ている。

Table 5.1: Si における平面波数と E_{el} の関係

A_{max}	E_{cut}	N_{pw}	NADIM	E_{el}	ΔE	CPU time
2.1	4.959	59	8	1.171122727	0e+0	2
2.6	7.601	113	12	1.043273992	0e+0	10
3.1	10.806	169	16	0.991663345	-8.8e-16	35
4.1	18.903	377	20	0.979594976	0e+0	53
5.1	29.249	725	24	0.974301525	-1.7e-15	106
6.1	41.843	1243	28	0.973366347	0e+0	200
7.1	56.687	1949	32	0.973159222	1.7e-15	365
8.1	73.780	2896	40	0.973127119	0e+0	658

このように、NADIM=16 では E_{el} は小数点以下2桁目が、NADIM=40 では E_{el} は小数点以下4桁目が収束することがわかる (Ry/cell 単位)。上の評価で使われた大胆な切り捨てを考えると、まあまあ予測一致といって良いだろう。式 (5.3) によると、相対誤差は h^2 に比例するので、NADIM を3倍に取ることで1桁精度が上がることになる。しかしそれと同時に、 N_{pw} は三乗で増えるので計算負荷の方がより早く増大するということになる²。

いずれにせよ、式 (5.3) に従うと、ポテンシャルの勾配が急になればなるほど、かつそこに電子密度集中していると (つまり電子がポテンシャルの谷間に局在していると) 離散化誤差は大きくなる。その時は平面波数に関する収束は遅くなることになる。

空間の有限分割による誤差はこれでわかったが、では例えば $E_{\text{cut}} = 10$ [Ry] くらいでは小数点2ケタ目以下の数字は信用できないのだろうか? 絶対値という点ではその通りだが、しかし前述し

²全エネルギー表式に現れる積分に対して、より精度の良い Simplon の公式を使うと少ない平面波でより収束が早くできる期待がある。平面波の増大に伴う計算負荷を考えると、やってみる価値がある。

たように多くの場合興味ある物理量はエネルギーの差である。有限分割によるポテンシャル分布、密度分布もそれらを与えられたものとして考えるなら、それらは誤差ではなくなる。つまり展開平面波の数を揃えるならば、エネルギーの相対比較はもっと下の桁まで可能である。その場合どの桁まで信用して良いのであろうか？それはこの計算の数値的安定性が決める。つまり(ii)の計算機の数値精度、丸め誤算に伴う誤差の伝播がどれくらい最終的な結果に影響するかということだ。

文献[29]を参考にすると、(簡単のため1変数関数で考え) 入力 x に対する最終結果 y は $y = \varphi(x)$ で計算されるが、入力 x の計算機精度による誤差(浮動小数点誤差)を相対誤差 ε_x で表すと、出力の相対誤差 ε_y は、

$$\varepsilon_y = \frac{x}{\varphi(x)} \frac{\partial \varphi(x)}{\partial x} \varepsilon_x \quad (5.4)$$

となり、式(5.4)の $(x/\varphi(x))\partial\varphi(x)/\partial x$ は誤差の増幅率を表し、条件数(condition number)と呼ばれている。この条件数を今の場合解析的に表すのは不可能だろうから、実験で評価してみる。具

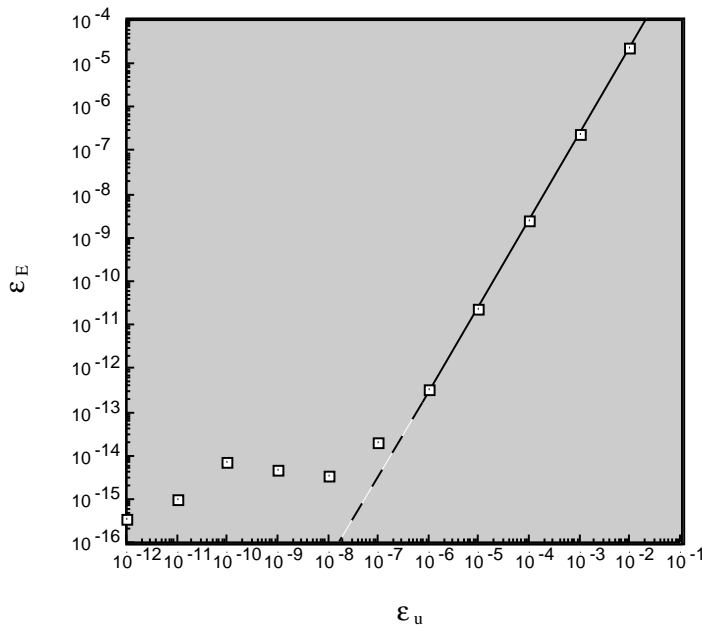


図 5.3: 原子位置に関する入力誤差 ε_u に対する、全エネルギーの出力誤差 ε_E

体的には、平面波数を固定し、原子配置 x をわずかに変えた系でのエネルギー比較 ($y = E_{el}$) をしてみる。Si₈ の結晶で、(0, 0, 0) 位置の原子を x 方向にわずかに動かしてみて E_{el} の変化を調べる。 $E_{cut} = 13.94$ [Ry] で $N_{pw} = 958$ と取る。その結果を図 5.3 に示す。この図からわかることは、位置のずれ u に対して非常に広範囲にわたり、条件数は 1 より十分小さく、数値的安定性があるということだ。位置の相対誤差 ε_u が 10^{-7} までの領域で全エネルギーについての相対誤差 ε_E との傾きから、 $\varepsilon_E \sim \varepsilon_u^2$ となることがわかる。これはもちろん平衡位置からのポテンシャル変化が調和近似になっていることの反映である。

こうして、pwm のプログラムは非常に数値的安定性が良いことがわかったので、条件さえ揃えておけば、先程の小数点 2 桁以下の数字でも十分意味があるということになる。但し今の Si の例は最も安定性の良い例と見るべきで、一般にはこれほどの安定性は望めない。だいたいにおいて収束性の悪いものは数値的安定性にも欠けると考えるべきだろう。

このエネルギー精度の問題は後で(6.1.3 節)また議論する。

5.2.3 再計算

一回の計算で十分な収束が達成されていないときは、今出力された波動関数を入力に戻して再度計算する。出力ファイル `pwm_*.wfn` を `inip_*.wfn` とリネームし、`pwm.para` 中の `iread` を 1 とセットする。これで再計算 OK である。

エネルギー評価

こうして得られた全エネルギーはどう評価したら良いだろうか。全エネルギーの定義は式 (1.22) で与えられているが、この式の中のポテンシャルは擬ポテンシャルで、電荷密度は価電子のものである。従って、これから評価されるエネルギーは結晶のエネルギーといっても、内殻電子の寄与は全く入っていない。それでも物質の化学的性質はほとんどが価電子で決まるという擬ポテンシャルの精神にのっとり、様々な興味あるエネルギーはこの全エネルギーの差で計算できることになる。

代表的なものは、凝集エネルギー、形成エネルギーなどである。固体 A の凝集エネルギー $E_{\text{coh}}(A)$ は

$$E_{\text{coh}}(A) = E(A^{\text{(gas)}}) - E(A^{\text{(sol)}}) \quad (5.5)$$

で、また化合物 $A_m B_n$ の形成エネルギー $E_{\text{form}}(A_m B_n)$ は

$$E_{\text{form}}(A_m B_n) = mE(A^{\text{(sol)}}) + nE(B^{\text{(sol)}}) - E(A_m B_n) \quad (5.6)$$

で計算できる。ただし、形成エネルギーの場合は、固体どうしの比較なので問題はないのだが、凝集エネルギーの場合は、孤立原子と固体との比較になるので、交換相関エネルギーの取り扱いに問題が生じる。この問題の取り扱いにはたくさんの議論があるが、ここでは擬ポテンシャルによる初期の文献 [28] を一つ引用するにとどめる。この問題は擬ポテンシャル固有の問題ではなく、LDA そのものの問題である。

5.2.4 電荷密度の表示

計算が無事終わったあと、`pwm` が出力する初めての物理量は上でみた全エネルギーである。しかし前節で述べたようにエネルギーの絶対値は意味がないので、この値だけからこの計算がもっともらしいものかどうかを判断することはほとんど不可能である（なんだ、さんざん計算した揚げく解ることはたったこれだけかとかっかりするかもしれません）。計算がもっともらしいものかどうかは、大抵の場合は電荷密度分布を見ることで判断することになると思う。であるので電荷密度分布を可視化することは是非必要だ。

実空間での全電荷密度の分布は `pwm_*.rho` に書き出される。データは、実格子の基本単位格子について、一方向について NADIM で一様間隔で取られたメッシュ点で記述される。

`pwm_*.rho` のフォーマットは、最初にディメンジョンとして、実空間でのメッシュ数 NADIM が書かれ、以下電荷密度 $\rho(\mathbf{r}) = \rho(i, j, k)$ が、 i が一番内側、 k が一番外側のループで書き出される。

このデータの可視化には、Mathematica のノートブック `ChgDnst.nb` が使われる。入力として、結晶データ `*.prim` と電荷密度分布データ `pwm_*.rho` を用いる。その使用例として、3次元の電荷密度をグラフィットの計算例で図 5.4 に示す。

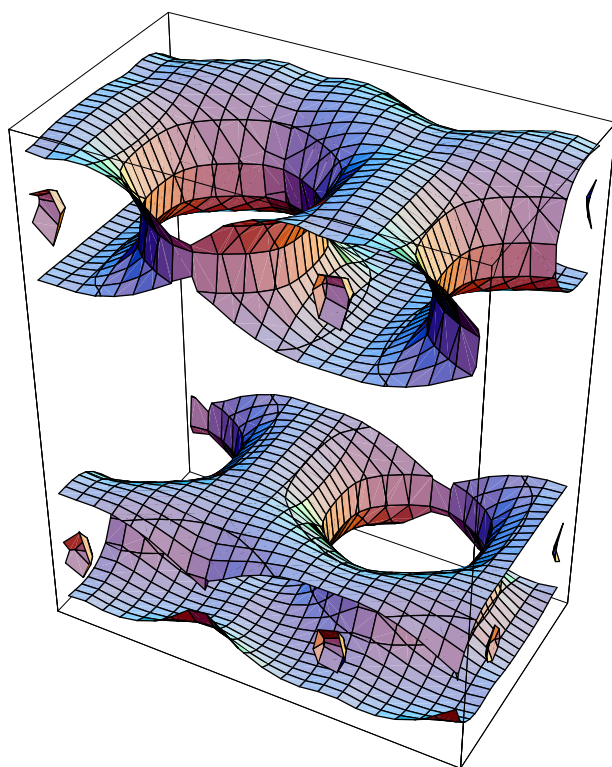


図 5.4: グラファイトの等電荷密度面

これから蜂の巣のボンドが層状にできていることがわかる。このように3次元の等電荷密度面図は差し当り電荷分布がどうなっているか素早く見たいとき有効である。

3次元の電荷密度は見掛け上アピールするものであるが、それに含まれる情報量はあまり細かくはない。むしろ2次元の切断面での等高線の方が結合の詳細を見るうえで必要である。その例が図 5.5 に示されている。今の入力データでは、 $z=1/4$ 面に2個のC原子がある。太い線が基本単位格子の境界で、グレーの点が原子である。等高線は黒い方が電子密度が高くなる。このように原子間に強い共有性結合ができていることが読み取れる。

もちろん、この2次元の切断面での等高線は電荷密度の詳細を与えかわり、文字通り全体の一面の情報しか与えられないので、それ一枚だけを見たのでは「群盲象を評す」類になりかねないということは常に念頭に置くべきだ。

なお、この部分の出力に関してだけは、その重要性に鑑み、Mathematicaのノートブック ChgDnst.nb の使い方を付録 A で説明する。

得られた電荷分布がもっともらしいかどうかの判断には、最終的には物理的素養が物を言う。という、「予測が付かないから計算に頼んでいるのであって、予測が付くくらいなら計算などしない」と言われそうである。かくいう作者自身も実はどれがもっともらしいのか判断に苦しむことが度々である。そうなると、その先は「素養」が違ってもやることは皆同じである。パラメータを変えたり、類似物をいろいろ調べたりと、そこからが物理が始まるという他ない。³

³類似物における電子分布の例をたくさん知っている人が、この世界では、所謂「専門家」と呼ばれ、少し

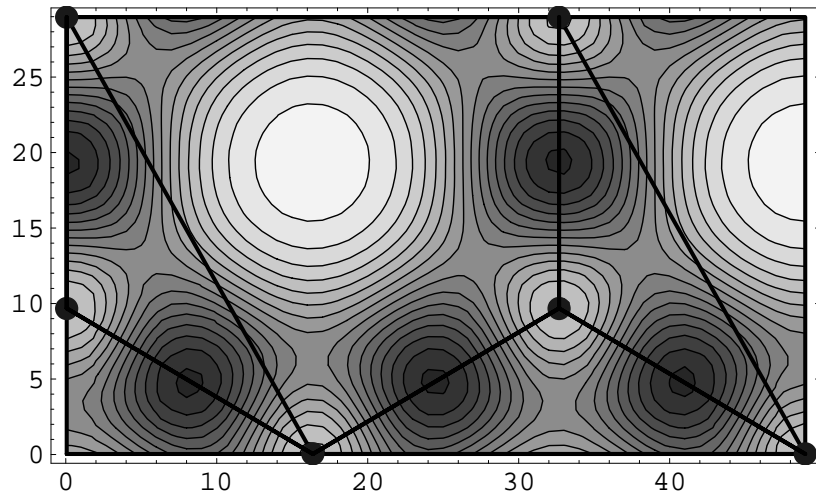


図 5.5: グラファイトの単位格子の $z=1/4$ 面での電荷密度分布

電子相関が効いている場合はこの方法では所詮限界があることも覚えておかなければならない。何しろ我々は $\sim 10^{23}$ 個の原子を相手にしているのだから。これは問題の解決に何の助けにもならないが、Callaway が密度汎関数法に関していみじくも述べている言葉を引用して締めくくる (文献 [32] の p. 106)。

これで (密度汎関数法のこと) 多体問題が解けたと思っはいけない。そのようなものは存在しない。もし誰かがそれを魔法のように成し遂げたとしたら、その時は物理は全くつまらないものになるであろう。

5.2.5 収束がうまくいかない場合

大抵の場合今述べた方法で十分収束するが、そうならない場合にも遭遇するかもしれない。奇数の電子数の場合それが起こり得る。この場合、価電子バンドのトップが電子占有数が他のバンドのものと異なるため、共役勾配法における次元最小化の過程で、放物線外挿法が成功する保証がなくなる。また本質的に準位間での準位交差が起こり収束が不安定になる要因になりうる。

似たような状況は金属で起きうる。金属の場合、価電子上端のバンドはゾーンのどこかでフェルミ面が横切るので、 k 点ごとにそのバンドの占有数が異なる。これが収束が安定化しない原因となりうる。

この場合の対処法はケースバイケースで万能の対処法は無い。個々のケースでの対処法に頼らざるを得ない。ゆえにそれらの対処法は詳細となるので、場合場合についてテクニカルシリーズを参照していただくとして、ここでは著者の経験からの基本的方針について述べる。

まず最初に認識して欲しいことは、電子数が奇数であるから、あるいは金属であるからしか知らない人が「素人」と呼ばれるのであろう。

という理由で必ず不安定になるということは無いということである。収束の悪かったことを何でも奇数個電子数のせいにはいけない。著者の経験では最高電子占有状態に縮退があるとき問題が起き、縮退が無ければ奇数電子数あるいは金属ということはほとんど問題とならない。

そこで収束に問題が生じたとき、まずそれがKS準位の縮退によって起きたものかを調べる必要がある。まず、インプットファイル `pwm.para` で

```
allow_posE ON
```

とオプション指定し、エネルギーが多少増加してもエネルギー最小化過程を強制的に進めて、エネルギー変化がどうなるかを見る。⁴

ところで「状態が縮退」という場合、以下の事項に注意を払う必要がある。

1. 縮退は対称性の理由により厳密に言うことができるものであるが、しかし数値計算の世界では数値だけが決め手である。数値には当然誤差があるので、縮退とはある数値範囲内での一致と理解される。
2. 一つの k 点内の状態間での縮退だけでなく、異なる k 点での状態間での縮退もある。これらは通常バンド理論では縮退しているとはいわないが、今の場合は縮退である。

より具体的には `pwm` ではバンドを一本一本更新しているが、その間は占有数は固定しているので、エネルギー的に近いバンドではその線形最小化の過程での放物線外挿法がバンドの順番を変えるかもしれないのである。放物線外挿法が失敗してストップした時点でそれが最低エネルギーを与えるものかどうか保証がない。

その他収束しない場合

以上述べた以外に収束がうまくゆかないことがあり得る。6.1.4節で詳しく議論するが、共役勾配方向を決めたあとの次元最小化が失敗する場合である。理論の1.7節で説明したが、次元最小化は式(1.45)について実行される。このとき全エネルギーの θ 依存性は式(1.46)のように、 2θ の関数で与えらるとしているが、これはあくまで近似でもっと高次項も存在する。多くの場合この近似は良い近似だが[18]、時々悪い場合があり、その場合は必ずしも収束しない。全エネルギーのほとんどの項は密度の線形近似で十分であるが、交換 - 相関項は密度の低次のべき乗では与えられないので、時として(特に平面波数が極端に少ない場合)この項の変化が主要項となり、上の近似が破綻することが起こる。

⁴というより、これによって初めてエネルギーが振動、発散したことがわかるのであり、そうしなければ振動したことさえも気づかないだろう。

第6章 電子の基底状態計算 (II)

pwm で電子系の収束が達成した後、pwm でできることにいくつかの選択ができる。

6.1 原子位置の最適化

結晶の全エネルギー E_{tot} は原子の配置 \mathbf{R}_i の関数とみなせる。基本単位格子に N 個の原子があれば $E_{\text{tot}}(\{\mathbf{R}_i\})$ は $3N$ 個の変数の関数となる。原子の平衡位置はこのエネルギー関数を最小にする配置で求まる。ただし原子位置に関するエネルギー関数はいくつもの局所的極小値があることが知られているので、こういう場合に大局的な最小値を求めることは困難か、あるいは不可能となる。

ここでは線形近似での局所的な極小値の探索を行うことを目的とする。従って、極小位置は原子位置の初期値に依存することを念頭に置くべきである。多変数関数の極小値探索のアルゴリズムはいろいろあるが、ここでは線形近似の¹共役勾配法を用いる。pwm では Hellmann-Feynman 力、すなわちエネルギー関数の勾配が計算できるので、この方法が特に有効である。

共役勾配法の適用により、基本単位格子内の全ての原子を一度に動かし、構造の最適化を図ることができる。しかしながら pwm では、原子の変位は与えられた結晶の対称性を破らない範囲内で与えられる。つまりワイコフ位置は変わらない。従って、たとえば空間群 O_h^7 の Si 結晶では、2つの原子が結晶の対称性により一意的に決められているので、この方法では全く動かないということになる。もちろん Si 結晶に対しても、対称性を落とせばこの方法が適用できる。

6.1.1 計算の準備

5.2.3 節のときの手続きと同様に前のセルフコンシステント計算の結果 `pwm_*.wfn` を用意し²、それから `pwm.para` を編集する。

```
Job (0:SCF Calc. 1:Atom optimize 2:Cell optimize 3:Phonon 4:MDS)
1
.....

number of iteration for electrons (maxIter0wfn)
7
.....
```

¹これは、共役勾配が決定された後の一次元最小化は放物線による外挿を用いることを意味する。

²絶対必要というわけではない。初めから原子位置を動かす場合は、初期波動関数としてあまり厳密に収束させるより、ラフな収束あるいは全くランダムな初期波動関数からスタートする方が全体としては計算効率は良い。

```

ftol
8.0D-5
.....

Resume atom relaxation
0
max number of iterations for atoms (maxIter0atom)
4

```

と Job Type の選択を 1 とする。原子位置に対する共役勾配法の繰り返し数の上限 maxIter0atom を適当な値にセットする。また 1 回の原子位置に対する波動関数の収束までの繰り返し数 maxIter0wfn は波動関数だけのセルフコンシステント計算のときのように多く取る必要はなく、だいたい上の値くらいでよいだろう。初期波動関数として、前の結果を使うのであれば iread を 1 とする。しかし波動関数データファイルは inip_*.wfn とリネームすることも忘れずに。これで pwm を走らせる。

計算は原子位置に関する繰り返し回数が maxIter0atom を超えるか、残留原子間力が ftol (Ry/a_B) 以下になるかで終わる。

一度の計算で十分な緩和が行われなかったときは Resume atom relaxation を 1 とセットすると、前の計算で達成した原子位置を初期条件としてもう一度最小化過程を始める。このときの原子位置情報は pwm_*.var により引き継がれる仕組みにしてあるので、*.prim まで書き換える必要はない。しかし波動関数は前の出力 pwm_*.wfn を inip_*.wfn とリネームする。

6.1.2 結果の読み方

ここで、わざと原子位置を次のように少しずらした Si₈ 結晶を例に取り、原子位置の最適化の計算を試みる。

POSITIONS RELATIVE TO A UNIT CONVENTIONAL CELL				TYPE	SYM(IG)	
1	0.0200000	0.0000000	0.0000000	1	si	1
2	0.0000000	0.5000000	0.5000000	1	si	1
3	0.5000000	0.0000000	0.4900000	1	si	1
4	0.5200000	0.4800000	0.0000000	1	si	1
5	0.2500000	0.2500000	0.2500000	1	si	1
6	0.7500000	0.7500000	0.2500000	1	si	1
7	0.7400000	0.2500000	0.7100000	1	si	1
8	0.2500000	0.7700000	0.7500000	1	si	1

このように相対座標が小数点以下 2 ケタ目で平衡位置からずれている。

これに 5 回の共役勾配過程をかけてみる。

STEP	Etot	delE	resid	rsf	frgr	dx/at
1	-63.22541398	0.0000E+00	0.2255E-08	0.2446E-01	-0.1957E+00	0.0000E+00
2	-63.27071131	-0.4530E-01	0.7378E-09	0.5797E-02	-0.4506E-01	0.5911E-01
3	-63.27328165	-0.2570E-02	0.1119E-10	0.2139E-02	-0.1607E-01	0.1425E-01
4	-63.27383294	-0.5513E-03	0.1495E-10	0.1819E-02	-0.1077E-01	0.8547E-02
5	-63.27454325	-0.7103E-03	0.1223E-10	0.1662E-02	-0.8373E-02	0.1648E-01

リストは、原子位置に関する共役勾配法のステップ数、全エネルギー (Etot)、エネルギー変化 (delE)、残留波動関数誤差 (resid)、残留 Hellmann-Feynman 力の平均値 (rsf)、各ステップでの共役勾配の方向の Hellmann-Feynman 力成分 (frgr)、原子あたりの移動量 (dx/at) の順に出力される。全エネルギー、残留力、それに伴い原子移動量が順調に減少していることがわかる。

その結果原子の相対座標は

```
Relative coordinates in a conventional cell
 1   0.0034588   0.0012806  -0.0060836   1   si   1
 2   0.0033400   0.5016982   0.4951692   1   si   1
 3   0.5013677  -0.0003884   0.4929211   1   si   1
 4   0.5052825   0.4977746  -0.0068789   1   si   1
 5   0.2536370   0.2497403   0.2445048   1   si   1
 6   0.7548898   0.7498467   0.2436450   1   si   1
 7   0.7533980   0.2489169   0.7411160   1   si   1
 8   0.2546224   0.7511310   0.7456130   1   si   1
```

のように、小数点以下3ケタ目まで平衡位置と一致していることがわかる。

ここで Hellmann-Feynman 力に関して注意したいのは、Hellmann-Feynman 力を単位格子内全原子について足しあわせたものは0となることである(つまり内力の和は0)。その結果、単位格子内の原子が動くときその重心は不変のままである。ということは、全原子を一様にずらしても良いということだ。もちろん0といっても、数値計算の場合は様々な要因で有限値となっている。であるので Hellmann-Feynman 力の和がどれくらい0からずれているかは計算精度のチェックになる。それはファイル pwm_*.frc に各原子に働く Hellmann-Feynman 力のリストのあと、

```
Check translational invariance
 0.4647E-05   -0.2650E-05   0.5672E-05
```

のように出力されるので、この部分を見ればよい。

6.1.3 力の誤差に関して

理想的にはこの極小値探索アルゴリズムにより残留 Hellmann-Feynman 力を任意の大きさまで小さくできる。しかし実際には、残留力がある値まで小さくなると、それ以上計算を繰り返してもそれ以上小さくならない。46 ページの「エネルギー精度」の項で議論したように、数値計算上の丸め誤差などが最終的にこの誤差に効いてきている。

原子間の力定数 k を仮定すると、平衡位置の付近でのエネルギー変化 ΔE は $\Delta E = \frac{1}{2}k(\Delta x)^2$ で近似できるので、 $f = k\Delta x$ より、力の誤差は

$$\Delta f = \sqrt{2k\Delta E} \quad (6.1)$$

と与えられる。従って力の誤差 Δf はエネルギー誤差 ΔE よりも悪くなる。問題は全エネルギーの誤差 ΔE をどう評価するかである。「エネルギー精度」の項で議論したように、有限平面波数からくる離散化誤差を主要な誤差と見ると、 ΔE は比較的大きくなるが、平面波数を揃えて計算機の数値精度からくるものだけを考えれば、 ΔE は非常に小さくなる。

実際には、個々の計算での残留誤差の起因を明らかにすることは大変難しいので、残留力の値が飽和してきた時点で（あるいは飽和しなくても計算が終わった時点で）、それが今の場合の数値誤差と見なすのが一番手っ取り早い。

残留 Hellmann-Feynman 力が Δf のとき、原子位置の不確定性 Δx は

$$\Delta x = \frac{\Delta f}{k} \quad (6.2)$$

で見積もらる。原子間の力定数 k はもちろん物によって違う。Si の場合であれば、動径方向の（すなわち伸縮）力定数は $k_r \simeq 0.3$ [Ry/Bohr²] くらいだ。それで残留 Hellmann-Feynman 力が 1×10^{-3} [Ry/Bohr] のとき、位置の不確定性は $\Delta x \simeq 3 \times 10^{-3}$ [Bohr] 程度ということになる。これは Si 結晶の相対座標では、小数点以下 4 ケタ目までの精度を意味する。一方それと垂直成分は、変角力定数 k_a だけが効いてくるとすると、変角力定数 k_a は k_r のせいぜい数%であるので、位置の不確定性は 100 倍くらい大きくなることになる。実際には結晶の場合、垂直成分が k_a だけで決まることはめったになく、別の方向からの k_r が効いてきて、結局値はそれらの中間になるだろう。したがって今の Si 結晶の場合は相対座標で、小数点以下 3 ケタ目までが保証されると考えればよい。

実際には、この Si₈ の場合は、「エネルギー精度」の項で見たように数値安定性が良い例で、先程の共役勾配法によって得られた原子配置から、再度同じ共役勾配法をかけると原子位置は 7 桁くらいまで精度を上げれる。しかしこの精度は他の物質では余り期待できないと思ったほうがよい。

6.1.4 収束に関する議論

全エネルギーの局所的極小の位置を求めるには共役勾配法は有効であることが知られているが、それでもいつでも成功するとは限らない。ここでは共役勾配法が失敗するいくつかの例を議論する。

理論のところでも述べたように、ここで用いている最小化法は、各ステップでの共役勾配による探索方向を決めた後、その方向での 1 次元最小化を行うのであるが、それは単純な放物線近似の最小化法を取っている。すなわちその方向ではエネルギー関数は $E(x) = ax^2 + bx + c$ と仮定される。既にある原子位置 $x = x_0$ （一般性を失わずそれを原点に取れる）でのエネルギー E_0 とその勾配（すなわち力） $f = -E'(0)$ がわかっているの、あと一点 $x = x_1$ で関数の値 E_1 がわかれば関数 $E(x)$ の形が決定できる。そして極小位置 x_{\min} は

$$x_{\min} = -\frac{b}{2a} \quad (6.3)$$

と決定される。ここに、 a と b はそれぞれ

$$\begin{aligned} b &= -f \\ a &= \frac{1}{x_1} \left(\frac{E_1 - E_0}{x_1} - b \right) \end{aligned} \quad (6.4)$$

で与えられる。従って、この方法は最小点を求める努力は最小で済むという利点を持っているが、その代償として放物線近似にしか適用できない³。幸い多くの場合、その近似はそ

³一般的な一次元関数の最小化のアルゴリズムはいろいろあるが、最小化にかなりの点の評価を取らねばならない。一次元での最小化に時間がかかりすぎると共役勾配法の利点は台なしになる。

んなに悪くないのだが、局所的極小が複数あることを反映して、この方法が適用できない場合が出てくる。

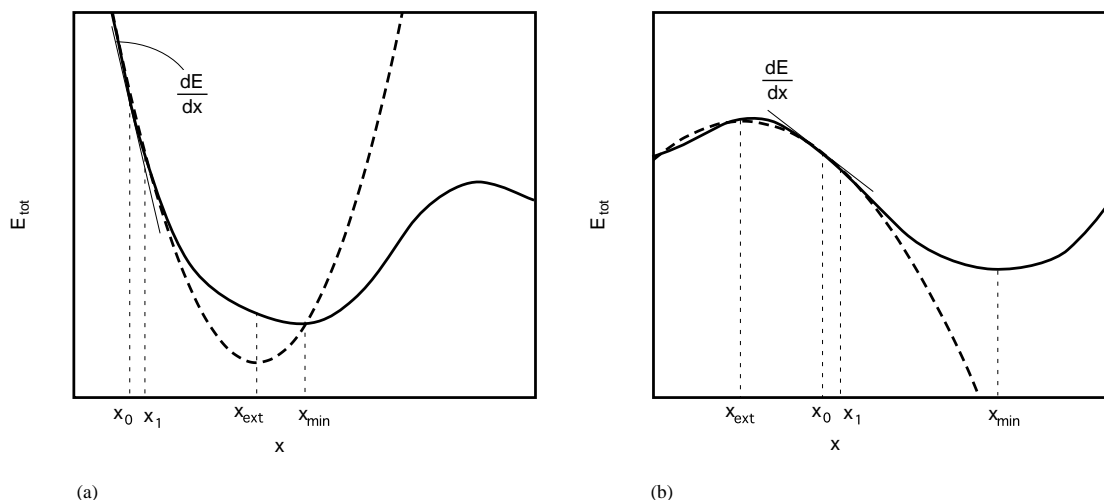


図 6.1: 放物線外挿が失敗する例。実線は真の断熱ポテンシャル。点線は放物線近似

例えば、断熱ポテンシャルが図 6.1(a) のようになっている場合がある。図からわかるように、 x_0 と $x = x_1$ からの外挿による極小点 x_{ext} は真の極小点 x_{min} から大きくずれている。それでも今度は x_{ext} を始点としてもう一度この線上で放物線近似の極小化をすればよさそうに見えるが、共役勾配法の特性として、一度探索した方向は二度と繰り返さないとという思想であるので、 x_{ext} に達したあと、次の探索方向はそれとだいたい直角方向で、この方向での極小値には達することはできない。

もう一つどうにもできない場合は、断熱ポテンシャルが図 6.1(b) のようになっている場合で始点 x_0 ともう一つの試行評価点 x_1 がたまたま図のようになっている場合だ。この場合はこの放物線は上に凸な曲線と理解してしまう。上に凸な放物線には最小値はないので、この放物線近似ではどうにもできない。この場合、プログラムでは、 x_1 を新たな始点として、かつその点から共役勾配法を始めからやり直すようにしてあるだけである。

そのようなときは、本質的にこの方法では解決できない。 x_0 に対する試行評価点 x_1 までの距離を大きく取るとかして、何とか放物線近似が良いところまでいかなければならない。

このような場合に出くわした場合ユーザーができる(すべき)ことは、まず、次元最小化過程が破綻したその方向に沿って断熱ポテンシャルがどうなっているか直接見ることであろう。それがもっとも確実な方法である。そのためにはソースコード `pw_relax.f` のサブルーチン `Atom_Optimize` を見る。そのサブルーチンの冒頭部分、変数宣言の部分に

```
LOGICAL :: step_flag=.FALSE. !step-by-step process
```

なる変数がある。この `step_flag` を `.TRUE.` とセットすると、`pwm` は原子位置に関する共役勾配過程を止め、与えられた原子変位方向に沿って、等間隔に点を取り全エネルギー変化を計算する。実際の計算は同じソースファイルのサブルーチン `Atom_Line_Step` で行わ

れるが、そのルーチンの中で、原子位置の単位変位、全ステップ数はそれぞれ `unit_dis`、`unit_max` で指定されているので、それを自分の目的に応じて設定すればよい。
結果は `pwm_*.sum` で

==== ION RELAXATION =====

```
...
STEP      Etot      delE      resid      rsf      frgr      dx/at
  1      -63.21392591      0.0000E+00      0.7951E-07      0.2444E-01      -0.1955E+00      0.0000E+00
```

の後、

```
0      0.000000      -0.63213926E+02      -0.19549217E+00
1      0.080000      -0.63228246E+02      -0.16235414E+00
2      0.160000      -0.63239896E+02      -0.12886472E+00
3      0.240000      -0.63248852E+02      -0.94986127E-01
4      0.320000      -0.63255078E+02      -0.60610679E-01
5      0.400000      -0.63258530E+02      -0.25618189E-01
```

とリストされる。各行は、順に、回数、変位の大きさ u 、全エネルギー E_{tot} 、探索の方向の Hellmann-Feynman 力成分 f_s となっている。 $u - E_{\text{tot}}$ の関係の微分が f_s を与えるので、そうなっているかもチェックする。

いずれにせよ、原子数が多くなっても、原子位置に対する共役勾配のステップ数 (`maxIter0atom`) はせいぜい 5 回くらいにしておくのが賢明である。

6.2 格子定数の最適化

原子位置に対する最適化と同様の原理で、`pwm` ではストレスが計算できるので、格子定数の最適化を計ることができる。格子定数はストレスが 0 となるように自己調整する。原子位置の最適化の場合の、力に相当するものがストレステンソル σ_{ij} で、変位に相当するものが歪みテンソル ϵ_{ij} である。

6.2.1 計算の準備

前の原子位置に関する最小化の時と同様に、初期波動関数 `inip_*.wfn` を用意し、`pwm.para` を編集する。

```
Job (0:SCF Calc. 1:Atom optimize 2:Cell optimize 3:Phonon 4:MDS)
2
atom movement (0: OFF, 1: ON valid only when cell optimization)
0
.....
Resume cell relaxation
0
max number of iterations for cell (maxIter0cell)
2
```

まず Job Type を 2 にセットし格子定数の最適化プロセスを起動する。このとき同時に原子位置も動かすかどうかの設定を次の行で行う。

格子定数に関する共役勾配プロセスの最大繰り返し回数を `maxIter0cell` にセットする。

計算は格子定数に関する繰り返し回数 `maxIter0cell` を超えるか、残留ストレスが `stress_tol` (デフォルト値 2×10^{-5} (a.u.)) 以下になるかすると終了する。

ここでは、簡単のためヘッダー IC に続く箇所のみを抽出してある。IC が格子定数に関する共役勾配のステップ数である。それに続いて、各ステップでの初期の全エネルギー (Etot0)、終点のエネルギー (Etot)、エネルギー変化 (del.E)、最小点の歪み位置 (epmin)、残留ストレスの大きさ (res stress)、その共役勾配方向の成分 (sea comp) が表示される。

ステップが進むにつれ、全エネルギー、残留ストレスが小さくなっていく様子がわかる。

この結果、格子定数は最終的に

```
The final lattice parameters
  9.38527485    9.38527485    9.38527485
  0.52019576    0.52019576    0.52019576
```

となる。

最後に、残留ストレスのリストが打ち出される。

```
total stress (Ry/Bohr^3)
  S(1,1)=    1.33098E-06    S(2,3)=    1.39142E-10
  S(2,2)=    1.33137E-06    S(3,1)=    2.91728E-18
  S(3,3)=    1.69774E-06    S(1,2)=   -1.92993E-18
```

初めのストレスと比べて2桁小さくなっていることがわかる。

6.2.3 収束に関する議論

格子定数の最適化における収束の問題は、原理的には原子位置の最適化におけるそれ (第 6.1.3、6.1.4 節) と同じである。何か問題の起きたときは、原子位置に関するときと同じように、探索方向での一次元での断熱ポテンシャルを調べることが一番確実である。

ソースコード pw_cellrelax.f のサブルーチン Cell_Optimize の中の変数宣言部分で

```
LOGICAL :: step_flag=.FALSE. !step-by-step process
```

という変数を見つける。この step_flag を TRUE. とすることで探索方向のスキャンを行う。実際のスキャンは同じファイルのサブルーチン Cell_StepByStep で行う。格子定数のステップ幅、ステップ数は、このルーチンの中でそれぞれ unit_step、nstep で与えられるので、適当な値に設定する。

結果は pwm_*.sum で見出し

```
==== LATTICE MODIFICATION =====
```

以下の部分を見る。

```
e=    0.00000000    Etot=   -15.80512513    ssd1=    0.00153912
e=    0.00200000    Etot=   -15.80511615    ssd1=   -0.01048361
e=    0.00400000    Etot=   -15.80508321    ssd1=   -0.02244878
e=    0.00600000    Etot=   -15.80502655    ssd1=   -0.03430808
e=    0.00800000    Etot=   -15.80494640    ssd1=   -0.04610712
e=    0.01000000    Etot=   -15.80484299    ssd1=   -0.05779727
```

と、歪み (e) に対する全エネルギー (Etot)、与えられた歪みの方向でのストレス成分 (ssd1) の変化をリストする。e-Etot をプロットしどういふふうに変化する

か、また `ssd1` は `e-Etot` の一次微分を与えるはずですので、そうなっているか確認する。

格子定数の精度は、すなわち歪みテンソル ϵ_{ij} の精度で与えられるが、弾性定数 C_{ijkl} を用いると、それは残留ストレス σ_{ij} と

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \quad (6.5)$$

の関係にある。誤差の評価のためには弾性定数のかわりに体積弾性率 (Bulk modulus) を用いれば十分だろう。

例として、先程の α ホウ素の場合の格子定数の数値精度を評価してみる。残留ストレスが $\Delta\sigma \approx 10^{-6}$ [Ry/Bohr³] である。体積弾性率 B は仮に Si と同じくらい $B = 100$ [GPa]、つまり 6.8×10^{-3} [Ry/Bohr³] とする。すると格子定数の数値精度は $\Delta\epsilon \approx 1.5 \times 10^{-4}$ 、つまり 0.01% くらいということになる。

平面波の選び方 格子定数を動かす場合、原子位置の最適化と違って特別の注意が必要である。それは平面波カットオフ半径についてである。同じカットオフ半径 k_c を用いても、格子定数が動くと、それに含まれる平面波の数が異なってくる。展開する平面波の数が異なってくると、エネルギー変化に不連続なジャンプが生じる [35]。明らかにこのような人為的なジャンプは 1 次元最小化を行うとき不都合となる。このため、`pwm` では格子定数の最適化を行うとき、最初与えた k_c によって選ばれた平面波のセットをそのまま通し続ける。これにより平面波数が増えることによる不連続点は避けることができる。一方、それは格子定数が変化するとき、正しい平面波の選び方をしていないという問題も持ち合わせている。

このあたりは、妥協を選んでいる。すなわち、歪みが小さいうちは、初期の平面波のセットを用いることによる誤差は無視できるであろうと考える。結果として歪みが大きく、最終的な格子定数が初期のものに比べて大きく異なるときは、改めてその点から計算をやり直すという考え方をする。ではどれくらい異なったときそうすべきかは、当然物、状況によるが、だいたい格子定数の変化で 10% くらいと思えば良いと思う。

6.2.4 圧力依存性の取り方

`pwm` はさらに静水圧下での格子定数の最適化も行う。 `pwm.para` で

```
pressure_GPa=  
10.0
```

とオプション設定をする。外部圧力は GPa 単位で入力する。正の値が加圧で、負の数字が減圧を意味する。

`pwm` は内部ストレスが、この外部圧力と同じになるまで共役勾配法を繰り返す。このオプションは、結晶構造の圧力依存性、相転移に応用できる。

第7章 電子構造スペクトル

セルフコンシステント (SFC) 計算が終了すると、バンド計算、DOS 計算などが可能となる。これらの計算を行う時 Brillian・ゾーンが具体的にどうなっているかの知識が必要で、例えば FCC の Brillian・ゾーンは図 7.1 のようになっていることを知らないといけない。それは文献 [24]などを参照していただく。Brillian・ゾーンを自分で描きたいのであれば Mathematica のノートブック MakeBZ.nb で書ける。

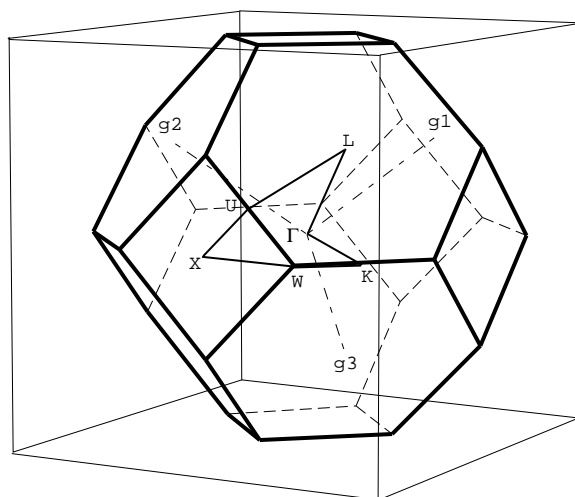


図 7.1: FCC の Brillian・ゾーン

バンド計算、DOS 計算などはプログラム pwbcd で計算する。pwbcd に関する入出力ファイルの関係は図 7.2 のようになっている。まずそれらの計算の元となっているセルフコンシステント電荷のデータ `pwm_*.rho` が必要である。 k 点サンプリングのデータ `inip_*.kpt` などは必要ないが、ポテンシャルデータ型を規定する `inip_*.inp` が必要となる。バンド計算、DOS 計算とも同じ入力ファイル `bcd.para` を使うが、中身が違うのでそれぞれ `band.para`、`dos.para` として作っておき、計算する時 `bcd.para` とリネームして使うのが便利である。

7.1 DOS 構造

7.1.1 準備

DOS 計算用の入力ファイル `bcd.para` は

```
JobType
dos
```

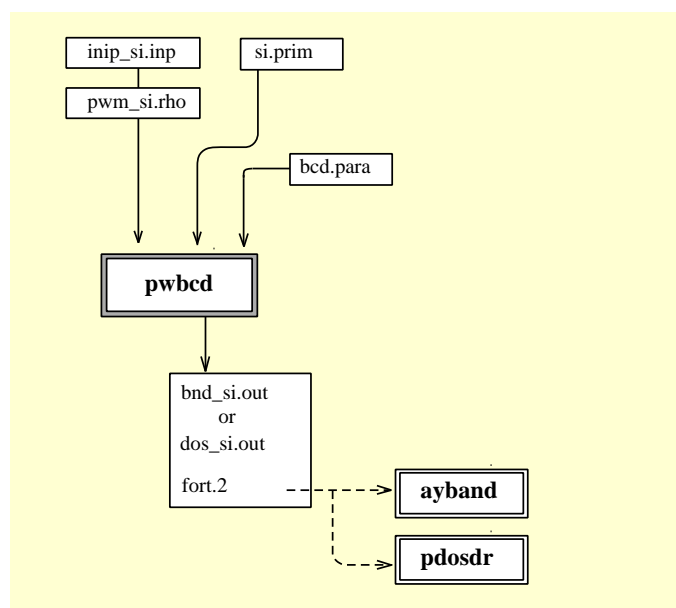



図 7.2: pwbcd 計算プログラムの流れ

```

Input file name
si.prim
number of division (nkdiv)
8
number of levels you want to draw (NBUP) usually NEPC
12
scan zone only (iscan)
1
print control (ilp)
1
use symmetry (isymm)
1
energy unit (ienun=0 for Ry, 1 for eV)
0
  
```

となる。

以下そこに現れるパラメータの意味を挙げる。

JobType	ジョブタイプ = dos
xtl.name	結晶名
nkdiv	k 線の分割数
NBUP	描くバンド数
iscan	デバッグのためのオプション。1 : デフォルト
ilp	プリントオプション 1 : デフォルト
isymm	対称化オプション 1 : 対称化する、0 : しない
ienun	エネルギー単位 0 : Ry、1 : eV

ここで一番重要な入力は DOS を計算するときのゾーンの分割数である。それを nkdiv に入力する。計算に取り入れるバンドの数 (NBUP) は平面波の数 (NHDIM) より小さ

ければ何でも良いのだが、典型的には価電子バンド数の2倍、つまり基本単位格子あたりの電子数を取ればよい。しかしアルカリ金属のように基本格子あたりの電子数が1とか2の場合は、あまりにもバンド数が少なくなるので、少し多めにとっておいたほうが良い。

その他のパラメータは主にデバッグ用であるので上の例のままが良い。

pwbcdはDOS計算結果をdos_*.tblあるいはfort.2に出力する。それをプロットするには、簡単には線スペクトルそのままのdos_*.tblをMathematicaのノートブックDOSshow.nbに掛ければ良い。より高度な処理として、既約表現ごとに分けられた線スペクトルデータfort.2をpdos(最新バージョンはpdosdr)に掛け、tetrahedron法[36]により滑らかなDOS曲線を得ることができる。なおpdosは柳瀬により作成されたものである。

DOS計算におけるk点の取り方は、基本逆格子 G_1 、 G_2 、 G_3 で作られる平行六面体を、一様にnkdiv分割する。例えばnkdiv=4とすると、 G_1 方向には G_1 単位で、

$$0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}$$

と4点取られることになる。Monkhorst-Packのkサンプリングの時と異なり、ゾーン中心から始まることに注意。それらのうちブリルアン・ゾーンの外にはみだたものは、自動的にその中に折り返される。また対称性で結ばれる点は消され、平行六面体は既約ゾーンに縮小される。従って最終的に決められたk点には重みがつくことになる。

dosclはそれらのk点、一点一点でハミルトニアンを対角化し固有値を下から並べて出力する。対角化の際、対称化オプションisymm=1としてあると、あらかじめ平面波は対称化される。

最後に全電子を下のレベルから順番に詰め、フェルミレベル E_F を割り出す。

SGI pwbcdでは、SGIマシンではコンパイラオプションを付けるとき、最適化レベルは基本レベル(-O)以上に上げないようにして欲しい。TSPACEがときに壊れることがある。また、-staticも付けたほうが安全だ。

7.1.2 DOS構造の結果

DOSスペクトルの結果はdos_*.tblあるいはfort.2に書き出される。その結果を見る前に、計算条件がファイルdos_*.outに書かれているので、まずそれをチェックする。見出し

```
===== The calculation parameters =====
```

の後には計算条件がリストされる。それに続いて見出し

```
===== K space segmentation =====
```

の後にはk点サンプリングに関する情報があるので、確認すること。上の入力の場合、初めに8点作られるが、対称性により次の3点

Number of sampling points (NKPTS)				3									
No	NM	c		p		DK	Nstr	WTK	in/out				
1	GM	0	0	0/	2	0	0	0/	4	0.0000	1	0.12500	1
2	L	-1	1	1/	2	2	0	0/	4	0.1688	4	0.50000	2
3	X	0	0	2/	2	2	2	0/	4	0.1949	3	0.37500	2
sum check over wtk =				1.000000									

にまで簡約される。 k 点の名前、conventional、primitive base の表現が与えられた後、その大きさ (DK)、星の数 (Nstr)、重み (WTK)、ゾーンの中か外かの指標 (TSPACE 参照) と並べられる。

その後、 k 点一点一点について、平面波を対称化して、既約表現ブロックごとに対角化する過程が記述される。前にも触れたが、pwm を始めとする「OSAKA2000」のプログラム群は基本的に primitive base で書かれており、対称化するために conventional base で書かれた TSPACE サブルーチンをコールしているため、その間の変換、逆変換が何度も挿入されている。それが正しく進行しているか途中経過のモニターをしている。一般ユーザーは無視してもらって結構である。

最後にフェルミレベル E_F を探索して締めくくる。

```

===== determin Ef =====
Determine the Fermi level
                                energy --- (eV)
Fermi energy =      6.2896
      at 1 th-kpoint      0 0 0/ 4
Total piled up number =      8.000000
      Number of electrons =      8
Sorting of HOMO
Valence Top:      6.2896 at 1      2 2 0/ 4
Sorting of LUMO
Conduction Bottom:      7.1785 at 3      0 0 0/ 4
Gap=      0.8889 from 0 0 0/ 4 to 2 2 0/ 4

final distribution of the valence bands
      kn  IP2 - IP3      E2 - E3      Occ2 - Occ3
1      3      3 - 4      3.17028 - 3.17028      2.00000 - 2.00000
2      2      3 - 4      4.91918 - 4.91918      2.00000 - 2.00000
3      1      2 - 4      6.28963 - 6.28963      2.00000 - 2.00000
Ef= 6.28963  nv= 3  kf= 1  del_n= 0.00000

Nel=      8.0000

Ef=      6.2896

```

刻まれたメッシュの分解能の中でエネルギーギャップも求められている。ギャップは 0.8889 [eV] でそれは、(000) の価電子帯上端から、(220)/4 の伝導帯下端までということが示されている。

次に価電子帯上端の電子占有状態が示される。Si では価電子帯は 4 本ある。従って縮退していなければ 4 番目のバンドが最上位占有バンドということになる。縮退がある場合は 4 本のうちいくつか最上位占有バンドと見なされる。上のリストで、IP2 から IP3 までのバンドが最上位占有バンドである。E2 と E3 はそれぞれのエネルギーレベルである。この例では全て完全にエネルギーが等しく厳密に縮退しているが、前述したように、エネルギーがぴったり一致しなくとも、差が ERESO 以内であれば縮退しているとみなされる。この IP2 から IP3 までのバンドは均等に電子が割り振られる。

また、電子がそれぞれの k 点の IP3 まで占有して全電子数が期待通り 8 個となっている

かを Nel でチェックする。最後にフェルミレベルの位置が出力される。

7.1.3 表示

DOS 構造を図示したい場合 2 つの選択がある。

(i) 線スペクトル

簡単には線スペクトルデータ dos_*.tbl をそのままプロットすることである。このファイルは

```
1    -5.885658    0.125000
1     6.289627    0.125000
...
2    -3.384971    0.500000
...
```

というように k 点の番号、エネルギーレベル、 k 点の重みをリストしている。Mathematica の DOSshow.nb はこのデータを読み、各線を適当な幅の Gaussian 分布に直し、全ての分布を重ね合わせて DOS を描く。その例を図 7.3 (a) に示す。

メッシュが荒いと見せかけの突起点がたくさん出るので、van Hove 特異点を探するときなどは、 k 点のメッシュをかなり細かく刻まないとわからないことが理解できる。¹

(ii) tetrahedron 法

tetrahedron 法 [36] はサンプリング点間のバンドを内挿して DOS 曲線を滑らかにする。pdos は doscl の出力ファイル fort.2 を読み込んでこのことを実行する。fort.2 のファイルフォーマットは 7.2.2 節で説明されるが、pdos で読み取られるように出力したものである。pdos はもともと別のバンドプログラム用だったので、DOS を書かせる考え方が若干違う。一番大きな違いは、いつものように pwbcd が primitive base の k 点サンプリングをしているのに対し、pdos は conventional base のものであることだ。両者が一致する場合は問題ないが、FCC 格子などの場合はユーザーがその間を調整する必要がある。²

先の Si の例だと、pwbcd では k 空間のメッシュを primitive base で 2 分割した。G₁、G₂、G₃ が直交軸でないため、conventional base での (000) と (001) の他、(111)/2 とその半分の点も現れた ($2\pi/a$ 単位)。一方、pdos は conventional base であるので、直交軸 a*、b*、c* を一様分割する。そのためここで分割数を 2 とすると、上の 3 点以外に、(100)/2、(110)/2 などが現れる。これらの点に関しては pwbcd はサンプリングしていないので当然エラーとなる。

従って pdos では分割数は半分の 1 とすべきである。この場合 pdos 側では、サンプリングとして、(000)/1、(100)/1、(110)/1、(111)/1 の 4 点と取る。対称性により、(111)/1 は (000)/1 と、(110)/1 は (100)/1 とそれぞれ等価となる。ということは pwbcd の 3 点のうち 2 点のみのデータを使い、(111)/2 のデータは無駄になるということだ。このあたりは不合理だがどうしようもない。

¹この点に関して柳瀬先生から御教示いただいた。

²この点では作者は全ての場合のチェックをしていないが、ここで述べた例に添ってプログラムをいじらず、入力パラメータのみを調整することで解決できていると思っている。もし問題が生じた場合は御一報を。

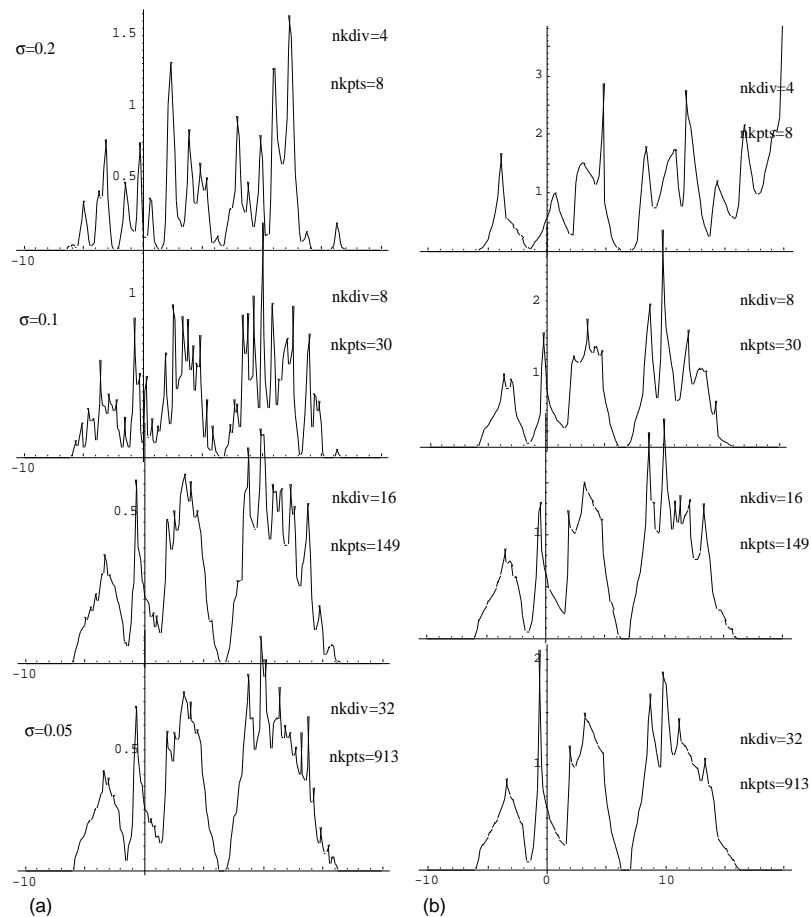


図 7.3: Si の DOS。(a) 線スペクトルを Gaussian でぼやかした例。(b) tetrahedron 法で滑らかにさせた例。

使い方は、まず入力ファイル pdos.para を以下のように用意する。

```
NONMSPIN-ORBIT
  1  1  1
0
-10.0 20.0 0.2
8
```

初めは磁性に関するもので、今の場合上のようにする。次の 3 つの数字が k 空間の分割数、NX、NY、NZ である。上で注意したように pwbcd の分割点と整合性を持たせます。次の数字は 0 とする。DOS を軌道による成分表示をするときに使う数字なので今は関係ありません。次に来る 3 つの数字は、エネルギーの下限 (ESTART)、上限 (ENEND)、一つのセグメントの幅 (DE) である。従ってエネルギースペクトルの点数は $(ENEND-ESTART)/DE+1$ で与えられる。最後の数字はセル内の全電子数である。

出力ファイル fort.12 は

```
SPIN 1
```

```

1 -10.0000 0.000 0.0000
2 -9.8000 0.000 0.0000

150 19.8000 1.689 27.1343
151 20.0000 1.438 27.4481
electron number= 8 EF= 6.800000000000000

```

のように最初と、最後のコメント行に挟まれ、線スペクトルデータがリストされる。各行はエネルギーメッシュの番号、エネルギー、DOS の値、そのエネルギーまでの DOS の積分値の順でリストされる。

このデータはやはり Mathematica のノートブック PDOSshow.nb で図示化できる。fort.12 の最初と最後のコメント行を外し PDOSshow.nb に持って行く。図 7.3 (b) に例が示される。メッシュ点の数が少ない場合も DOS 曲線は滑らかになることがわかる。それでも van Hove 特異点を正確に求めるにはやはり多くのメッシュ点が必要であることには変わらない。

7.1.4 フェルミ面

金属の場合は、pwbcdd の出力からフェルミ面を描くことができる。図 7.1 にあるように、pwbcdd は出力ファイル dos_*.sum を出力するが、これはファイル fort.2 と違い、既約表現ごとではなく k 点ごとのバンドレベルを出力する。これを Mathematica ノートブック MakeFE.nb に読み込ませ、また dos_*.out に出力されるフェルミレベルを入力することでフェルミ面を描く。もっともこの作画プログラムはフェルミレベルを単なる入力パラメータとして扱っているの、フェルミレベルでなくとも、与えられたエネルギーでの等エネルギー面を描くことにも使える。

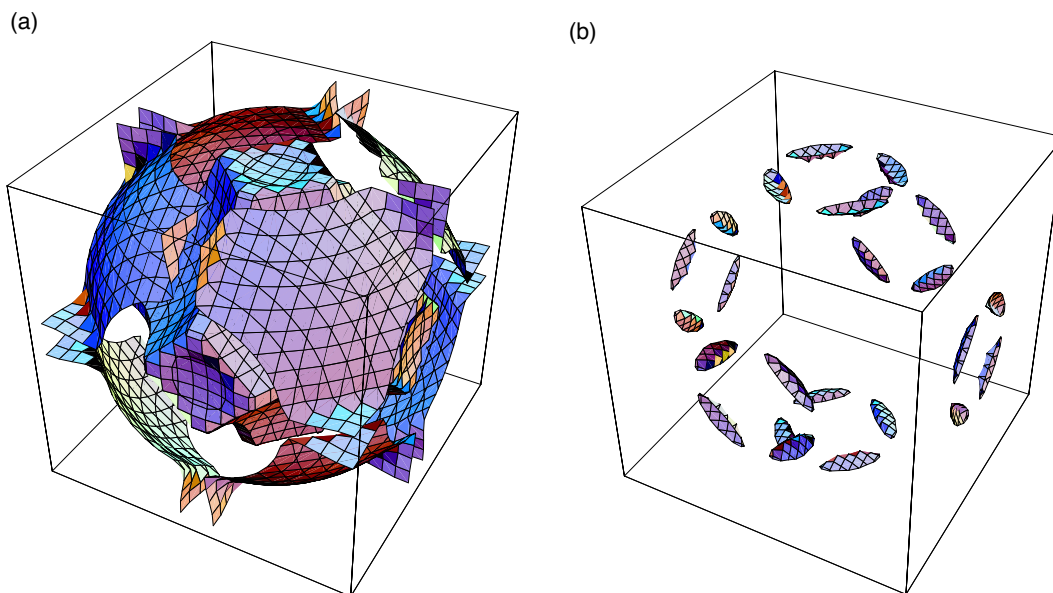


図 7.4: Al のフェルミ面。a) バンド 2。(b) バンド 3。

図 7.4 には、Al のフェルミ面の例がでている。表示は、第一ブリルアン・ゾーンにだけ

でなく、拡張ゾーンで描いた。作画には少し時間がかかるが、しょっちゅう使うのでなければ我慢できる。本当はこのフェルミ面に図 7.1 のようなブリルアン・ゾーンが重ね書きされるとわかり易いのだが、Mathematica の使い方の問題で今のところそれはうまくいっていない。図 7.1 のブリルアン・ゾーンと比較してみると、バンド 2 はゾーンの内と外で、球形の殻でなっていることがわかる。

7.2 バンド構造

band は指定された k 空間上の線（多くは対称線）に沿ってバンドを計算する。それらの線は一筆書きされるように取る。結果の表示には DOS の場合と同じく、2 通りの仕方を用意している。ひとつには、 k 点に対して対称性を抜きにしてバンドレベルを書き並べたファイル band_*.tbl を使うもので、それは簡単な XY プロットングプログラムで図示化することができる。もう一つがプログラム ayband を用いたより高度な処理で、バンドを滑らかに描くことができる。ayband はその出力を PostScript ファイルとして出力する。これで Illustrator などを読むことができる。

なお ayband は柳瀬章により作成されたものである。

7.2.1 準備

バンド計算では bcd.para は

```

JobType
bnd
Input file name
si.prim
number of k points specifying symmetry lines (NKPTS)
7
KB(3), ICB      (in prim)
0   0   0   1   G
3   3   6   8   K
1   1   2   2   X
1   2   3   4   W
1   1   1   2   L
0   0   0   1   G
0   1   1   2   X
number of division per line (NDIV)
5
number of levels you want to draw (NBUP) usually NEPC
12
scan zone only (iscan)
1
print control (ilp)
1
use symmetry (isymm)
1

```

のようになる。

以下そこに現れるパラメータの意味を挙げる。

ここで一番重要な入力にはバンドを描く対称線の指定である。pwbcd はバンドを描くときゾーンを一筆書きすることを前提としている。したがって、描く径路は

$$A \rightarrow B \rightarrow C \dots$$

JobType	ジョブタイプ = bnd
xtl.name	結晶名
NKPTS	バンドを描くときの対称線の数
KB, ICB	対称線の両端の k 点
NDIV	各対称線上の分割数
NBUP	描くバンド数
iscan	デバグのためのオプション 1 : デフォルト
ilp	プリントオプション 1 : デフォルト
isymm	対称化オプション 1 : 対称化する、0 : しない

となり、ノードの数が NKPTS で、以下ノードの k 点が KB(3)/ICB に primitive base で与えられることになる。この例では、 $L-\Gamma$ 、 $\Gamma-X$ 、 $X-W$ 、 $W-K$ 、 $K-\Gamma$ の 5 本の対称線が取られることになる。³NDIV には一本一本の対称線の分割数を入れる。

NBUP に描きたいバンドの数を指定する。DOS を計算するときと同様に、単位格子あたりの電子数くらいが良いが、それが 1 とか 2 くらいのときは少し多めに取る。ここでどのバンドまで描くかについて述べなければならない。DOS のときもそうだが、use symmetry を 1 にセットすると、各 k 点で、各対称ブロック/em ごとにバンドを NBUP だけ計算し、積み上げる。であるので NBUP が少ないと、ある対称線から別の対称線に移ったときエネルギーが上のバンドが消えることがある。その時はこの NBUP を増やす。

後のパラメータはデバグ用なので、上の値のままで良い。

7.2.2 バンド構造の結果

pwbcd は指定された線に沿ってバンドを計算し、ファイル bnd_*.tbl および fort.2 に結果を出力する。しかし結果を見る前に DOS 計算のときと同じく、計算条件が正しく反映されているか、bnd_*.out で確かめておくのが良い。

手っ取り早くバンド図を見たければ、bnd_*.tbl を二次元プロットングプログラムにかけるのが良い。そのフォーマットは

```

1   -0.248790244660327
1   -6.646822029304370E-002
...
2   -0.289100855937076
...
```

というように、 k 点の番号の後、エネルギーがリストされるのでプロットングは容易である。その例は図 7.5 に示される。横軸は $L-\Gamma-X-W-K-\Gamma$ の順で各線が 5 分割でプロットされている。対称性に関する情報が欠けているだけでなく、点の集まりとして表されているので隣の点同士のつながりは人間の手で書かなければならない。今の例ではそれはそう難しくないが、バンド数が密集してくると横のつながりは簡単ではなくなる。

³柳瀬先生によると、もっと賢い取り方(対称線を全て通る)は、 $\Gamma \rightarrow K \rightarrow X(1,1,0) \rightarrow W \rightarrow L \rightarrow \Gamma \rightarrow X(1,0,0)$ と取ることである。

ゾーン境界でのバンドの曲り具合も微妙で、メッシュ点の数を多くしてもなかなか難しいものである。

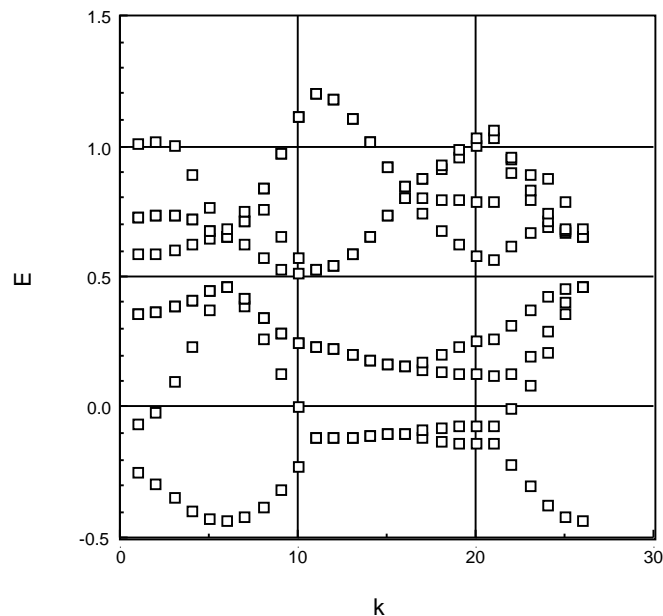


図 7.5: S_i のバンド図。band の出力点をそのままプロットしているだけ。

こうした問題は、バンドの対称性、ゾーン境界における異なる表現間の可換関係を利用すると解決される。これを行うのが ayband である。ayband は既約表現に従って各バンドを spline 曲線で滑らかにつなぐ。またゾーン境界での曲りにも気を配っており、それが k について 1 次かあるいは 2 次になるかを、わざわざゾーン境界を越えた点まで延長して調べている。

同じ既約表現ブロックに属しているバンドは「基本的」には交わらない(準位の不交差)であるので ayband は、例えば一般点ではバンドが如何に近づこうが絶対交わらないように分散曲線を引く。プログラムとしてこの処置には、著者は不満はないが、概念的にそれが絶対正しいのか著者は今一つ確信が持てない。2 つの縮退した準位は、その間に如何に小さくとも相互作用があれば、その縮退は解ける。しかし同じ既約表現ブロックに 3 つ以上の準位があるときでも絶対、縮退がないのかというと著者は確信を持っていない。「偶然縮退」というものがあるのも良いと思うのだが。「偶然縮退」については、過去いろんな人が議論しているが、例えば Herring はゾーン内に「偶然縮退があったとしたら」それはどういう状況になるか詳しく論じている [37]。しかしそれは「偶然縮退があった時」そうなるというもので、あるかどうかは問題にしていけない。ともかくも ayband で描く限り「偶然縮退」は存在しないことになる。

ayband の入力ファイルとして k 点での小群の既約表現ごとにバンドレベルを分類したものがファイル fort.2 である。

データは k 点ごと、一つの k 点については、その小群の既約表現ごとにブロックに分けられ出力される。その一つのブロックは次のようになる。

```
1   5 5 5 10 1 L 2 1 1 5
   2.793676   5.247479   6.851224   7.409376   8.271489
```

始めの数字は通し番号で、次の4つの数字で k 点が表されている。この例では (5, 5, 5)/10、つまり L 点となるが、この表現は conventional base のものである。次がスピンの関係した制御数 (IUD) で、 k 点の名前が続く。

残りの4つの数字は、順に既約表現番号 (MRN)、表現の次元 (MWET)、 k の星の数 (NST)、このブロックに属する固有値の数 (NEIG) となっている。

そして改行して、個々の固有値が小さい順に NEIG 個だけ並べられる。

この fort.2 を使って ayband はバンド図を描く。

ayband はもともと汎用のバンド図作成プログラムである。そのため様々な作図指示用のパラメータがある。それがファイル fort.3 に書き込まれる。その中身は

```
NONMSPIN-ORBIT
0 0 50                      NLCOMP NSPIN IFILE
0 1 12                      JPR     JMARK IPOINT(character)
-1.0 1.2 100.0 150.0
  5
  4 4 4 8 0 0 0 8 LD
  0 0 0 8 8 0 0 8 DT
  8 0 0 8 8 4 0 8 W
  8 4 0 8 6 6 0 8 K
  6 6 0 8 0 0 0 8 SM
0.5                          Fermi level
Si PseudoPotential
```

となる。上から順番に

MAGNET 磁性状態。pwm では NONMSPIN-ORBIT のみ有効。

NLCOMP, NSPIN, IFILE 角運動量による成分分解の数。スピン状態。これらはいずれも 0 とする。IFILE は出力ファイルの番号

IPR, JMARK, IPOINT 最初の2つは出力制御パラメータ。上の例のようにする。IPOINT はバンド図の中の表現ラベルの文字の大きさ。

E0, EM, YM, XM プロットするときの、エネルギー最小値、最大値、 y 軸、 x 軸スケールの大きさ (mm 単位)。

NAXM 次の描こうとする k 空間上の線の数

線の指定 線の端点 k_1 、 k_2 により指定。 k は3つの分子とその後の共通分子で表される。以下これが NAXM まで繰り返される。端点の名前は便宜上のもので読まれない。

EF フェルミレベル

TITLE 図のタイトル

このように ayband では改めて、pwbcd とは独立にバンドを描く線を指定する。

図 7.6 にはこのようにして得られたバンド図を示す。

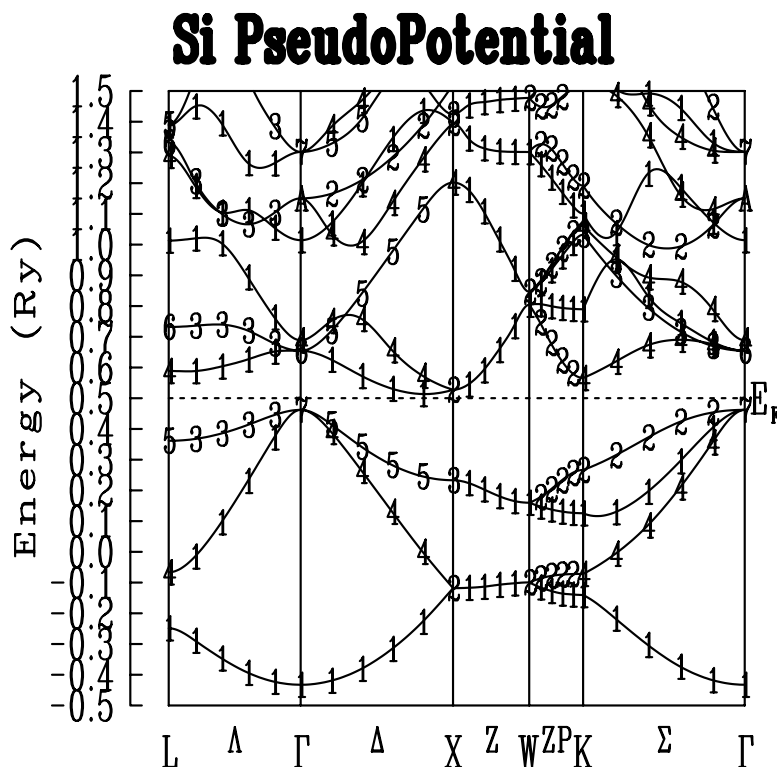


図 7.6: ayband を用いた Si のバンド図。k 点間の長さも正しく与えられていることに注意。

このようにバンド線がスムーズに描かれ、ゾーン境界でのつながりも明確になった。

図中の番号は既約表現の番号であるが、それは TSPACE が機械的にかけてに付けたものである。従って、よく文献で見かける既約表現のラベルと比較する時、その対応を調べる必要がある。

ayband は時間反転に関して縮退するものは、一方の既約表現の番号のみしか表示しない。例えば、閃亜鉛型の GaAs などでは (T_d^2)、 Δ 軸上の既約表現 3 番と、4 番のバンドは時間反転対称性のためくつつくが、ayband はラベルとして 3 のみを付ける。したがってバンド数、縮退度を見るときこの点注意が必要である。

このとき話をややこしくしているのが、文献の指標表が、通常の既約表現 (ベクトル表現) のものと、射線表現 (ray, projective, or multiplier representation) のものがあることだ。著者はどちらかという、古い文献がベクトル表現で、新しいものが射線表現になっている印象を受ける⁴。TSPACE は射線表現になっている。作者の印象としては、固体のバンド計算論文ではベクトル表現のラベルが用いられるほうが多いと思うので (多分

⁴ベクトル表現の例が Koster[38] だが、230 の空間群全てを尽くしているわけではない。全てを網羅して

Koster のもの [38] が一番参照されているのではないかと思う) それとの対応について説明する。

TSPACE は既約表現の完全なリストを出力することができる (サブルーチン TSIRDS)。前述しましたように TSPACE の出力する表現は射線表現 $D(\alpha)$ である。それは元の、 k の小群のベクトル既約表現 $\Gamma(\alpha)$ に、位相因子を掛けたもの

$$D(\alpha) = \exp(ik \cdot \tau_\alpha)\Gamma(\alpha) \quad (7.1)$$

となっている。ここに τ_α は回転 α に付随する格子周期を持たない並進 (nonprimitive translation) である。この $D(\alpha)$ 表現は、ゾーン内の対称線上では nonprimitive translation に伴う余計な位相因子が取り除かれ、点群の表現との対応が取り易くなるが、ゾーン境界では混乱の原因となり得る。

非共型 (nonsymmorphic) 空間群の具体例 Si で示すと、 Δ 軸上では、サブルーチン TSIRDS は

```

IMR NO 1 DIMENSION= 1
  1 2192227284042
  1 1 + + + + + + + +
IMR NO 2 DIMENSION= 1
  1 2192227284042
  1 1 + + + + - - - -
IMR NO 3 DIMENSION= 1
  1 2192227284042
  1 1 + + - - + + - -
IMR NO 4 DIMENSION= 1
  1 2192227284042
  1 1 + + - - - - + +
IMR NO 5 DIMENSION= 2
  1 2192227284042
  1 1 + - I J 0 0 0 0
  2 2 + - J I 0 0 0 0

```

と出力する (ファイル fort.15)。この表では小群に含まれる全ての要素に対しての行列要素がリストされる (見やすいように対角成分のみ抽出している)。この中の記号 (要素番号や指標の値) は TSPACE[24] 参照のこと。この射線表現の指標をクラスごとに見やすくしたものが表 7.1 の左側である。

これを見ると、射線表現 $D(\alpha)$ では、点群 C_{4v} の表現との対応が明白となる。一方通常の表現 $\Gamma(\alpha)$ では、半端な複素数 η が指標に現れる。 $k \rightarrow 0$ の極限を取ればこれは 1 となるので点群 C_{4v} の表現との対応を取るのにそんなに困難はないが、半端な複素数が現れない分、射線表現の方がすっきりしている。

しかし、非共型のゾーン境界では話はそう簡単ではない。表現が次元の場合では点群と等価 (射影等価) となり点群の表現が使えるが、射線表現では指標に複素数が現れ通常の点群の指標表との対応が取りにくくなる。

2 次元以上の表現ではもはや対応する点群は存在しない。多分一番複雑なのは X 点だろうから、それを示す。その射影表現は

いるものは、射線表現のもので、Kovalev[39]、Bradley Cracknell[40] によるものがある。しかし後者は完全とはいえ、初心者 (著者も含まれる) が、その中から目的とする既約表現表を取りだすのは相当の忍耐が必要だ。そしてせっかく得られた既約表現のラベルも結局 TSPACE と同じく単なる機械的数字にしか過ぎない。少なくとも著者には Mulliken や BSW 記法の方が親しみ易い。

Table 7.1: Δ 軸上の既約表現指標表。左が射線表現 D 、右がベクトル表現 Γ のもの。二行目が回転に付随する非基本格子ベクトル $\tau = (1/4, 1/4, 1/4)$ を示す。左の第一列目が TSPACE の既約表現番号を示す。

D		E	C_2	$2C_4$	$2\sigma_v$	$2\sigma_d$	Γ	E	C_2	$2C_4$	$2\sigma_v$	$2\sigma_d$
				τ	τ			τ	τ			
1	Δ_1	1	1	1	1	1	Δ_1	1	1	η	η	1
2	Δ'_1	1	1	1	-1	-1	Δ'_1	1	1	η	$-\eta$	-1
3	Δ_2	1	1	-1	1	-1	Δ_2	1	1	$-\eta$	η	-1
4	Δ'_2	1	1	-1	-1	1	Δ'_2	1	1	$-\eta$	$-\eta$	1
5	Δ_5	2	-2	0	0	0	Δ_5	2	-2	0	0	0

$\eta = \exp[-i\mathbf{k} \cdot \tau]$

```

IMR NO 1 DIMENSION= 2
  1 2 3 4161819222526272840424346
  1 1 + + + + 0 0 0 0 0 0 0 0 - - - -
  2 2 + + - - 0 0 0 0 0 0 0 0 - - + +
IMR NO 2 DIMENSION= 2
  1 2 3 4161819222526272840424346
  1 1 + + + + 0 0 0 0 0 0 0 0 + + + +
  2 2 + + - - 0 0 0 0 0 0 0 0 + + - -
IMR NO 3 DIMENSION= 2
  1 2 3 4161819222526272840424346
  1 1 + - - - J I J I 0 0 0 0 0 0 0 0
  2 2 + - - - J I I J 0 0 0 0 0 0 0 0
IMR NO 4 DIMENSION= 2
  1 2 3 4161819222526272840424346
  1 1 + - - - I J I J 0 0 0 0 0 0 0 0
  2 2 + - - - I J J I 0 0 0 0 0 0 0 0

```

とでる。

前と同じように、指標表として書き並べると、表 7.2 の左側のようになる。

Table 7.2: X 点の既約表現指標表。左が射線表現 D 、右がベクトル表現 Γ のもの。指標が全て 0 となる要素は省略してある。左の第一列目が TSPACE の既約表現番号を示す。

D		E	C_{2x}	$2C'_2$	$2\sigma_d$	Γ	E	C_{2x}	$2C'_2$	$2\sigma_d$
				τ			τ			
2	X_1	2	2	0	2	X_1	2	2	0	2
1	X_2	2	2	0	-2	X_2	2	2	0	-2
4	X_3	2	-2	$2i$	0	X_3	2	-2	2	0
3	X_4	2	-2	$-2i$	0	X_4	2	-2	-2	0

前述したように、Koster[38] などの表では通常の既約表現しかでていないので (表 7.2 の右側) 式 (7.1) により通常の表現 $\Gamma(\alpha)$ に戻した指標を pwbcd の出力ファイル fort.18 に書き込んだ。そこには各点の既約表現の指標がリストされている。

```

11 1 th point = 40 0 0/ 40 X 2 1.0000 0.0000 0.0000

Characters
#IR= 1 ND= 2 MG= 16
JG= 1 2 3 4 16 18 19 22 25 26 27 28
Re= 2.00 2.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

```

```
Im= 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
JG= 40 42 43 46
Re= -2.00 -2.00 0.00 0.00
Im= 0.00 0.00 0.00 0.00
```

このように点群操作の番号とともにその指標が実部と虚部に別れて出力される。それと文献の指標表と照らし合わせて通常のラベルに直せばよい。この作業（群の要素番号の参照と指標の読み込み）はいささか疲れるが、そこまでやれば他人が読める立派なバンド図となる。

7.3 波動関数表示

第8章 分子動力学法

第一原理分子動力学法とは、断熱近似のもと、原子の動きを Newton の運動方程式に従いシミュレーションするものである。第一原理と形容詞がつくところは、原子に働く Hellman-Feynman 力を電子状態から求めることを意味する。この力を第一原理から求めるのに膨大な時間がかかるのであるが、Car Parrinello 法の出現により、その部分の計算時間が大幅に短縮されるようになった [20]。

元の Car Parrinello 法は、波動関数 ψ_n と原子位置 \mathbf{u}_κ を独立変数と見なし、それらの変数、およびさらにあれば付加的な自由度 α に関する Car Parrinello ラグランジュアンを立て、それから導き出された Euler-Lagrangian 方程式

$$\mu \frac{d^2 \psi_n}{dt^2} = -\mathcal{H} \psi_n + \sum_m \Lambda_{nm} \psi_m \quad (8.1a)$$

$$M_\kappa \frac{d^2 \mathbf{u}_\kappa}{dt^2} = \mathbf{f}_\kappa \quad (8.1b)$$

$$\mu_\alpha \frac{d^2 \alpha}{dt^2} = -\frac{\partial E_{\text{tot}}}{\partial \alpha} \quad (8.1c)$$

を時間発展方程式として解くものである。ここに μ や μ_α はそれぞれ、 ψ_n 、 α に付随した仮想的質量で、 Λ_{nm} は波動関数の直行規格化に関係するラグランジュアンの未定乗数である。波動関数に関する時間発展方程式は、形の上では最急勾配の方向に ψ_n が変化するものとなっている。しかしこのプログラムでは、 ψ_n は時間発展で計算するのではなく、これまでの通り共役勾配法で静的に求めている。各時間ステップの間、波動関数は共役勾配法で基底状態に持っていきが、しかしそれはあまり厳密である必要はないというところがこの方法の利点である。

第一原理分子動力学シミュレーション (MDS) が行うことは、原子に適当な初期速度を与え、その後の原子の運動を追跡することである。ところで有限温度 MDS における温度 T とは何か？それは原子の運動エネルギーの平均値である。すなわち、

$$\frac{1}{N_{\text{at}}} \sum_{\kappa} \frac{1}{2} M_\kappa v_\kappa^2 = \frac{3}{2} kT \quad (8.2)$$

で T が与えられる。この式 (8.2) はもっともそうに見えるが、しかし実際の MDS においてはいくつかの注意が必要である。まず、式 (8.2) は古典近似なので、低温では成り立たなくなる。それと本来の統計力学上の温度の定義では温度は運動エネルギーの時間平均を取るべきものである。それに対して式 (8.2) での運動エネルギーは瞬時瞬時のエネルギーである。

このようにして温度 T が決められると、様々な有限温度の現象がシミュレートできるようになる。アニール処理は典型的な有限温度シミュレーションの応用例である。それはま

た、原子位置の大局的な平衡位置を見つけることに利用できる。また相転移などにも応用できる。また多数の原子の運動の時間変化を追跡しているの、それから速度や密度相関関数などが計算でき、従って様々な線形応答（フォノンや原子の自己拡散係数など）がわかる、などさまざまな応用が考えられる。

ここで単位について触れておく必要がある。原子単位では通常

$$\frac{\hbar}{E_h} = \frac{ma_B^2}{\hbar} = 2.42 \times 10^{-17} \quad (\text{sec}) \quad (8.3)$$

で定義される時間を原子時間単位とする（それをここでは Hartree sec と呼ぶことにする）。しかしここでは、エネルギー単位は Ry であるので、それに従って、時間単位も定義（8.3）の2倍をとり、それを Rydberg sec と呼ぶことにする。

それに伴い、この原子単位で Newton の運動方程式は、

$$\Delta^{(2)}X = \frac{F}{M_{\text{at}}} \frac{2m}{M_u} (\Delta t)^2 \quad (8.4)$$

となる。 M_u は原子質量単位 ($1.6605402 \times 10^{-30}$ [g])、 M_{at} は原子質量である。原子単位では、 $m = 1/2$ であるので、 $M_u/2m = M_u$ 、すなわち電子質量の半分を基準にした原子質量となる。

Newton の運動方程式は数値計算では、時間に関する2次の差分方程式（8.4）として解かれるが、その際効果的な方法として、Verlet のアルゴリズム [41] が採用されている。すなわち、

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + \frac{f(t)}{M} \Delta t^2 \quad (8.5)$$

$$v(t) = \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t} \quad (8.6)$$

である。このアルゴリズムは簡単でありながら、高次の精度を保ち、かつ時間発展に対する安定性も良好という点でとりわけ秀逸なものである。

8.1 シミュレーションの準備

第一原理分子動力学シミュレーションを行うということは、基本的に原子を自由に動かすので（特別の場合は原子変位に対称性などの拘束をつけるということもあるが）、ここでのプログラム pwm で取り扱えるのは対称性のない結晶（つまり P1 群）のみということになる。

この場合も初期波動関数の入力原理上必要ないが、このプログラムはそれを前提とする。従って動力学シミュレーションの前にあらかじめ Job Type=0 で pwm を動かし、（きっちり収束している必要はないが）初期波動関数を作っておかなければならない。だいたい第一原理分子動力学シミュレーションは計算時間が長いので、その計算時間を見積もるうえでも一度 pwm を動かしておく方が見通しがつく。この初期波動関数データも含

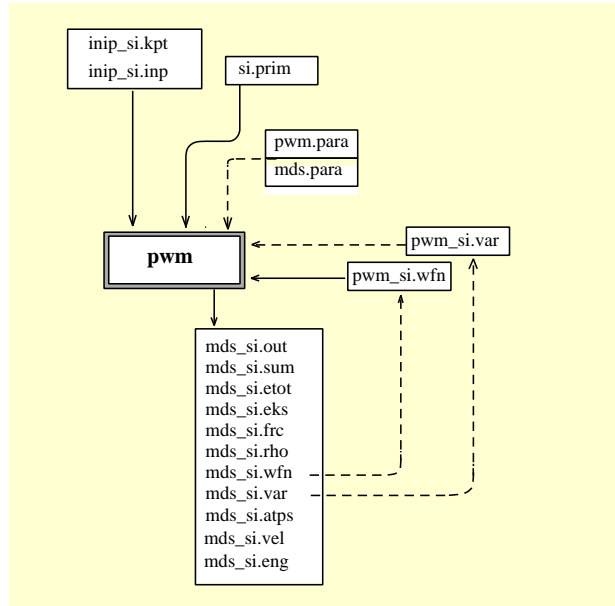


図 8.1: MDS に関するファイル

め、mds のまわりのファイルを図 8.1 に示す。

まず、pwm.para で

```
Job (0:SCF Calc. 1:Atom optimize 2:Cell optimize 3:Phonon 4:MDS)
4
```

と Job Type を 4 とセットし MDS を起動する。このときの波動関数に関する収束条件はそのまま有効である。原子位置の最適化のときと同じように波動関数の収束までの繰り返し数 maxIter0wfn は少なめでよい (~ 7 くらい)。

次に MDS に固有の入力パラメータはファイル mds.para に書く。

```
time step (TMDL) in Ry sec
25.
number of iterations for atom move (MaxIter0mds)
4
mds_Resume
0
way of giving velocity (0:zero, 1:random, 2:given) (ignored in case mds_Resume=1)
1
Initial kinetic energy of atoms in K (valid only for ivcond=1)
1200.
Temperature program (0: OFF, 1: velocity scaling, 2: Nose-T, 3: Nose-TP)
0
```

以下にそれぞれのパラメータの意味を順に説明する。

時間ステップ (TMDL) は Rysec 単位で与えるが、系の原子の典型的応答時間より短く取るべきだ。すなわちだいたい最高振動数のフォノンの周期より短く取る。全体のシミュ

TMDL	時間ステップ (Rysec 単位)
MaxIter0mds	MDSの最大ステップ数
mds_resume	MDSの再計算 0:しない、 1:する
wayOfGiveV	原子初期速度の与え方 0:初期速度0、 1:ランダム、 2:手入力
Tat_inp	原子初期運動エネルギー 単位 K
tp0ctrl	温度プログラム 0:温度制御なし、 1:速度スケーリング、 2:Noseの一定温度制御、 3:一定温度・圧力制御

レーション時間 Tspan は

$$Tspan = MaxIter0mds * TMDL \quad (8.7)$$

となる。Tspanの逆数がこのシミュレーションでの振動数の分解能を決める。

原子の初期速度として、3通りの与え方を用意している。変数 wayOfGiveV の値によって、

- 0 全速度を0に設定。
- 1 ランダムに速度を与える。このときのみ初期原子温度 Tat_inp が読まれ、乱数で与えられた速度の絶対値をその温度で規格化する。
- 2 全速度を手動で入力。この場合は、ファイル vatom_inp から原子ごとの速度を3成分ごとに与える。

という選択がある。

分子動力学シミュレーションは計算を一旦終わった後、さらにそれを引き続ける必要がしばしば生じる。このようなときは、計算出力 mds_*.wfn を pwm_*.wfn と変更し、mds_Resume を1と設定すると前の計算の続きを実行する。このとき、原子の位置および速度のデータは mds_*.var をリネームした pwm_*.var から読まれるので、mds.param で設定される初期温度や原子速度は無視される。

8.2 シミュレーションの結果

trivial な例として、Siの基本単位格子を対称性をなしとした結晶を取り上げる(結晶名 rhsi)。すなわち基本単位格子は、簡易単位格子と同じで、菱面体をしていて、 $N_{at} = 2$ である。k点サンプリングは $(\pm 1, \pm 1, \pm 1)/4$ の中からの4点、 $E_{cut} = 10.80$ [Ry]、 $N_{pw} = 169$ を取る。

この結晶に対して、まず初期原子の温度を600 [K]と与え、時間ステップとして100 [Ry·sec]をとり、50ステップまでシミュレーションしてみる。その結果は、mds_rhsi.sumで見る。いつものように計算条件が与えたものと同じとなっているか確認する。

特に、その中で、初期原子速度の項を見る。

```

Ion temp =          600.0 (K)   =          0.00380 (Ry)
  initial velocities in a.u.
  No          v (au)          mag
  1   -0.1280E-03  0.5663E-03 -0.3293E-03   0.6675E-03
  2    0.1280E-03 -0.5663E-03  0.3293E-03   0.6675E-03
  Ion energy=    0.011405 (Ry)   =    600.000 (K)

```

初めの、Ion temp と出ているのは kT のことで、mds はそのエネルギーを各原子の運動エネルギーに割り振り（ただし速度方向はランダムに選ぶ）かつ全運動量が 0 となるように調節する。最後に Ion energy として $(1/2)Mv^2$ の平均が出力するので、式 (8.2) のように $(3/2)kT$ となっていることを確認する。

その後、

```

===== MOLECULAR DYNAMICAL SIMULATION =====
Maximum iterations (IHF)= 50
TIME STEP          100.000 (Ry sec)
  corresponding freq. =          206.70 (THz)

```

の見出しの元で、MaxIter0mds や時間ステップ Δt が書かれている。corresponding freq. というのは、ちょうど Δt を 1 周期とする周波数を [THz] 単位で表している。

続いてシミュレーションの結果は、

STEP	ETTL	Etot	Ekion	=> T (K)	resid	rsfs	dx/at
0	-15.24045955	-15.25186451	0.1140E-01	600.00	0.303E-10	0.303E-06	0.000E+00
1	-15.24025576	-15.24943273	0.9177E-02	482.79	0.265E-10	0.249E-01	0.183E+00
2	-15.24000484	-15.24447379	0.4469E-02	235.11	0.318E-10	0.409E-01	0.245E+00
3	-15.23986732	-15.24067524	0.8079E-03	42.50	0.241E-10	0.483E-01	0.274E+00
4	-15.23986373	-15.24019961	0.3359E-03	17.67	0.365E-10	0.485E-01	0.278E+00
5	-15.23996980	-15.24313559	0.3166E-02	166.55	0.192E-10	0.426E-01	0.257E+00
6	-15.24016456	-15.24780201	0.7637E-02	401.80	0.258E-10	0.307E-01	0.210E+00
7	-15.24037874	-15.25122632	0.1085E-01	570.68	0.475E-10	0.135E-01	0.129E+00
8	-15.24042165	-15.25043339	0.1001E-01	526.70	0.397E-10	0.205E-01	0.157E+00
9	-15.24013282	-15.24528059	0.5148E-02	270.82	0.366E-10	0.470E-01	0.227E+00
10	-15.23976098	-15.24035866	0.5977E-03	31.44	0.219E-10	0.646E-01	0.258E+00
...							
48	-15.24044369	-15.25111029	0.1067E-01	561.16	0.392E-10	0.147E-01	0.134E+00
49	-15.24025722	-15.24710186	0.6845E-02	360.09	0.444E-10	0.398E-01	0.209E+00
50	-15.23984274	-15.24150023	0.1657E-02	87.20	0.222E-10	0.618E-01	0.251E+00

と打ち出される。

これまでの意味での全結晶エネルギー Etot、原子の運動エネルギー Ekion、そしてそれらを足しあわせた ETTL がリストされる (Ry 単位)。Ekion は [K] でも表されている。そして、残留波動関数誤差 resid、残留力 rsfs、原子あたりの移動量 dx/at が続いてリストされる。エネルギーの時間変化などをプロットしてみると良い。Ekion が 0.011 [Ry] の振幅で振動していることがわかる。それに応じて結晶エネルギー Etot が (運動エネルギーに対するポテンシャルエネルギーの役を果たす) やはり振幅 0.011 [Ry] で、位相を 180° 変えて振動している。その結果全エネルギー ETTL は小数点以下 3 ケタ目まで一定になっていることがわかる。完全に一定でないのは波動関数の収束が達成されていないからだ。

なお、原子の運動エネルギーを温度換算した Ekion [K] をプロットしてみるとわかるが、それは 600 [K] の初期値を与えられ、その後 600 [K] から 0 [K] の間を振動していることがわかる。従って原子の運動エネルギーの時間平均は 300 [K] と、入力値の半分となっている。初期の原子位置は平衡位置で、すなわちポテンシャルエネルギーの最小値なので、時

間平均すると、入力運動エネルギーの半分はポテンシャルエネルギーの増加に使われたと理解できる。すなわちビリアル定理に従っていることがわかる。

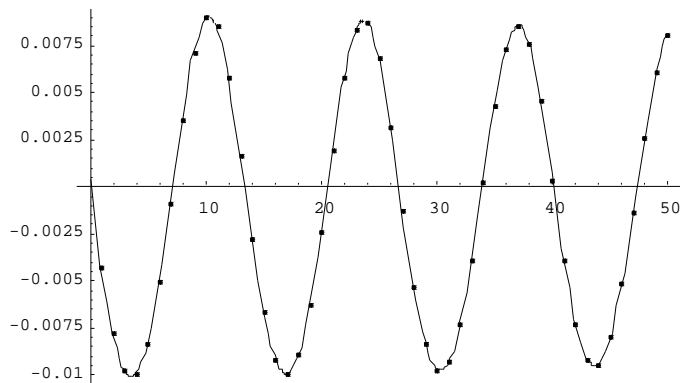


図 8.2: 初期原子運動エネルギー $T = 600$ [K] を与えたときの Si 原子 1 の x 方向運動。時間ステップは 100 [Rysec]

このようにいろいろな物理量の時間変化をプロットしてみることは重要だ。ここでは、原子位置の時間変化についてプロットしてみる。各時間ステップの原子位置はファイル mds_rhsi.atps に書き込まれる。

```
===== MOLECULAR DYNAMICAL SIMULATION =====
          POSITIONS OF ATOMS ( A,B,C, AS UNIT )
          ATOM      A          B          C
```

の見出しの後、

```
ITER :    1
      1  -0.0044176    0.0108640   -0.0055570
      2   0.2544176    0.2391360    0.2555571
ITER :    2
      1  -0.0079128    0.0195239   -0.0098936
      2   0.2579129    0.2304761    0.2598936
...

```

と各時間ステップの原子の相対座標が出力される。原子 1 の位置を時間 (x 座標) の関数としてプロットしたものが、図 8.2 に出ている。きれいな単振動をしていることがわかる。これから周期は 13.3 ステップである。すなわちこれは振動数にして

$$\frac{206.70 \text{ [THz]}}{13.3} = 15.5 \text{ [THz]} = 512 \text{ [cm}^{-1}] \quad (8.8)$$

となり、Si の光学モードの振動数と一致する。

またこのときの振幅をチェックしてみる。温度 T の時の平均振幅 \bar{u} は、

$$M\bar{u}^2\omega^2 = 3 \left(\bar{n} + \frac{1}{2} \right) \hbar\omega \quad (8.9)$$

$$\cong 3kT \quad (8.9a)$$

で与えられる。ここに \bar{n} はボーズ・アインシュタインの占有数である。近似式 (8.9a) は高温で成り立つ。 T を [K] 単位で、周期 τ を原子時間単位 [Rysec] で表したとき、式 (8.9a) より、平均振幅 \bar{u} は原子単位で

$$\bar{u} = \frac{\tau}{\sqrt{M_{\text{at}}}} \times 2.30 \times 10^{-5} \sqrt{T} \quad (8.10)$$

と評価できる。

この結晶 rhsi は、単位格子が直交軸上にないことに注意して、運動の 3 成分ごと図 8.2 のように振幅を調べ、平衡位置からの移動距離の平均を求めると、 $\bar{u} = 0.0933783$ [Bohr] となる。周期 τ は 13.3×100 [Rysec] だから、 $T = 300$ [K] とすると、式 (8.10) から、 $\bar{u} = 0.099$ [Bohr] となり一致することがわかる。

8.2.1 有限温度シミュレーション

ここで、今終了した結果を保存し、波動関数ファイル mds_rhsi.wfn を pwm_rhsi.wfn とリネームし、それを初期波動関数として再度シミュレーションを続けてみる。今度は温度を制御してみる。

入力パラメータファイル mds.para で、mds_Resume を 1 とセットし原子運動を再開することを宣言する。

全ステップ数 MaxIter0mds を 50 とし、その間の温度プログラムを設定する。

```
Temperature program (1: ON, 0: OFF)
1
keep period (itkeep), Tstart, Tend, itend
5, 300.0, 150.0, 45
```

これは、初めからステップ 5 まで、初期温度を 300 [K] に保ち、それからステップ 45 まで温度を 150 [K] まで連続して、リニアに下げる。ステップ 45 以降は温度は 150 [K] のまま保つという意味である。

ここでの温度制御は、式 (8.2) で定義される瞬時瞬時の原子の運動エネルギー T_k と、制御温度 T_{ref} により、各原子の速度を一様に

$$\sqrt{\frac{T_{\text{ref}}}{T_k}} \mathbf{v}_k \longrightarrow \mathbf{v}'_k \quad (8.11)$$

と加減することで行う。

まず、mds_rhsi.sum で与えた温度条件をチェックする。

```
Temperature Control: ON
  hold      T =   300.0 (K)      until step   5
  change
    from T =   300.0 (K)      at    6
    to   T =   150.0 (K)      at   45
  rate   -3.85E+00 (K/step)
```

と冷却速度が計算される。

そのあと、結果が打ち出される。

STEP	ETTL	Etot	Ekion	=> T (K)	resid	rsfs	dx/at
...							
2	-15.23926026	-15.24496274	0.5702E-02	300.00	0.312E-10	0.499E-01	0.217E+00
3	-15.23390437	-15.23960685	0.5702E-02	300.00	0.133E-10	0.680E-01	0.253E+00
4	-15.22448951	-15.23019199	0.5702E-02	300.00	0.297E-10	0.888E-01	0.301E+00
5	-15.22528774	-15.23099022	0.5702E-02	300.00	0.283E-10	0.794E-01	0.310E+00
6	-15.23201130	-15.23771378	0.5702E-02	300.00	0.431E-10	0.553E-01	0.296E+00
7	-15.23587092	-15.24150029	0.5629E-02	296.15	0.180E-10	0.460E-01	0.278E+00
8	-15.24066198	-15.24621824	0.5556E-02	292.31	0.255E-10	0.366E-01	0.243E+00
9	-15.24310791	-15.24859107	0.5483E-02	288.46	0.295E-10	0.297E-01	0.216E+00
...							
40	-15.24771892	-15.25093570	0.3217E-02	169.23	0.324E-10	0.163E-01	0.170E+00
41	-15.24846891	-15.25161258	0.3144E-02	165.38	0.188E-10	0.838E-02	0.132E+00
42	-15.24797924	-15.25104981	0.3071E-02	161.54	0.262E-10	0.158E-01	0.109E+00
43	-15.24680696	-15.24980442	0.2997E-02	157.69	0.208E-10	0.248E-01	0.146E+00
44	-15.24338337	-15.24630772	0.2924E-02	153.85	0.232E-10	0.398E-01	0.200E+00
45	-15.24093444	-15.24378568	0.2851E-02	150.00	0.187E-10	0.479E-01	0.224E+00
46	-15.23717009	-15.24002133	0.2851E-02	150.00	0.201E-10	0.616E-01	0.249E+00
...							

と原子の運動エネルギー温度で見ると、期待通りの温度履歴となっている。

ところで、このときの原子の運動はどうなっているだろうか？図 8.3 (a) にやはりこの

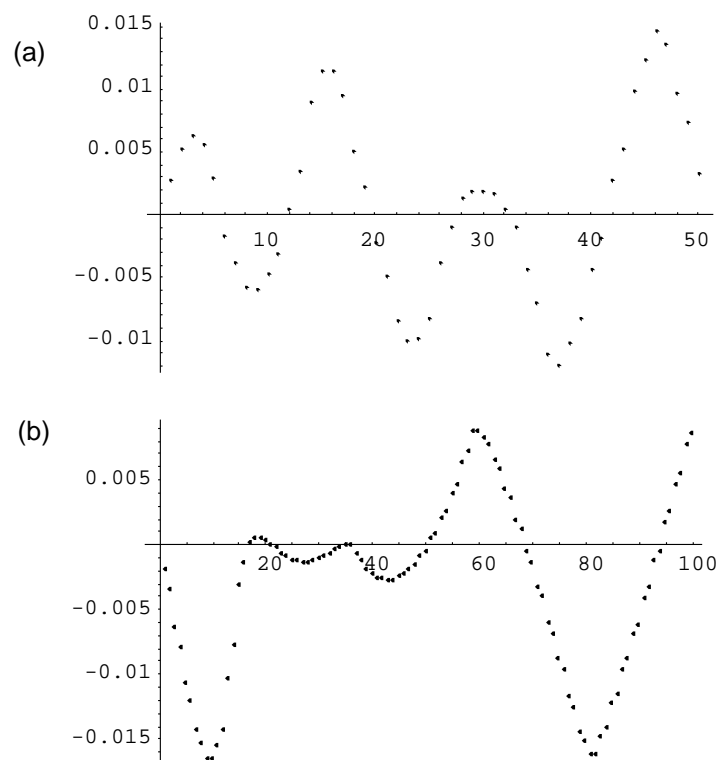


図 8.3: rhsi の $T = 300$ から 150 [K] までの温度シミュレーション。原子 1 の x 座標の変化。(a) 時間ステップ 100 [Rysec]。(b) 時間ステップ 50 [Rysec]。

ときの原子 1 の x 座標の変化を示される。直感的には、減衰振動をしていることが期待されるが、一見して全く違ったものとなっている。次に、温度履歴はそのまま、時間ステップを細かく刻んでみる。同図 (b) には同じ全時間スパンだが、ステップ時間 Δt を半分の 50 [Rysec]、全ステップ MaxIter0mtds を倍の 100 で取ってみた結果である。これまた期待と異なるだけでなく、(a) とともに明らかに異なる。単に時間ステップを細かくしただけなのに運動が全然違ったものになる。いったいどれが正しいのだろうか？

こうなった理由は温度の定義を思い起こせば難しいことではない。MDS では各時間の原子の運動エネルギーは、式 (8.11) に従い、プログラムされた温度に強制的に合わせられるようにしている。本来温度 T の熱浴に浸されている系では、各原子のミクロな運動は図 8.2 で示されるように、常に揺らいでいる。今の場合はたった 2 粒子なので、揺らぎというより周期的振動になっているが、 N 粒子のエネルギー E は

$$\frac{\Delta E}{E} = \sqrt{\frac{2}{3N}} \quad (8.12)$$

で揺らいでいる。温度とはもともとマクロには定常状態での、そのようなミクロの揺らぎの時間平均で与えられているものである。ところが、式 (8.11) では揺らいでいる量そのものを温度と定義しているのをおかしなことになったのだ。「揺らいでいる温度」とは本来の温度の定義に反する。揺らいでいるのはエネルギーの方であるべきなのだ。

このようなわけで、この MDS は非現実的な仮定 (8.11) を持っていて、特にそれが少数粒子系の場合 (たった 2 粒子!)、極端な形で現れることになる。各瞬間、瞬間の速度を、その「瞬間の温度」でスケールするということは、その瞬間粒子に散乱を与えるのと同じ効果をもたらすので、時間刻みを細かくすれば、それだけ散乱が多くなり、全体として全く異なる運動となるわけだ。エネルギーを人為的に吸い取った瞬間に、それは熱擾乱の働きをし、もはや個々の運動を決定論的に追及することは放棄しなければならない。

それでは我々はいったい何をシミュレートしているのだろうか？このような状況下では、図 8.3 の (a) (b) とともにアンサンブルとしてはそれなりに正しい。そして物理的意味を持つのは、それらの時間平均だけということだ。いちいち個々の粒子の軌跡については止めるべきである。

有限温度の多粒子系を扱うどんなシミュレーションでも、結局のところ、統計量、平均量だけが意味を持つ。速度スケーリングの仮定 (8.11) をすることで、ミクロな各瞬間瞬間の温度をコントロールする非現実性に我慢できないなら、いくつかのより優れた有限温度シミュレーションのアルゴリズムが存在する。能瀬の方法 [42] などが代表的なものである。

多粒子系への適用

以上述べた速度スケーリングの問題も、粒子数が多くなるとその非現実さは和らぐものと期待される。それでこの分子動力学法が、多粒子に応用されたときどれくらい現実をシミュレートするものであるかを、粒子数を増やした Si_8 の結晶で見てみる。

k 点サンプリングはやはり $(\pm 1, \pm 1, \pm 1)/4$ の中からの 4 点¹、 $E_{\text{cut}} = 18.90$ [Ry]、 $N_{\text{pw}} = 1503$ である。

¹ Si_8 の結晶では、 Γ 点のみの 1 点サンプリングでは不十分だ。

この結晶に対して、初期温度を $T = 600$ で与えて、その後の運動を見たものが図 8.4 に出ている。今度は運動エネルギーの変化は単なる単振動よりは複雑になっている (図 8.4a)。

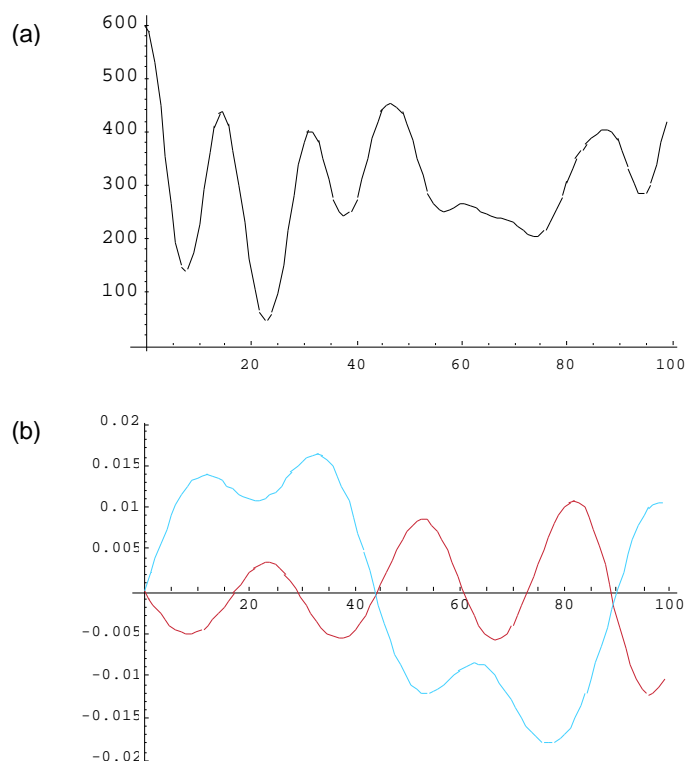


図 8.4: (a) 初期値 $T = 600$ を与えた Si_8 の運動エネルギーの時間変化。時間ステップ 50 [Ry·sec]。 (b) 原子 1 の x 座標 (赤線) y 座標 (青線) の変化。

それで各粒子の運動を分解してみると、同図 (b) を見ると、 x 座標の動きは式 (8.8) で求めた Si のゾーン中心光学モードにほぼ近似できるが、 y 座標の動きはそれにほぼ倍の周期のモードが重なっていることがわかる。つまり、プリミティブな Si 結晶を Si_8 に拡張したため、プリミティブな Si 結晶のゾーン境界モード (それはおおまかにはゾーン中心モードの振動数の半分) が現れてきたためである。

このとき、原子の平衡位置からのずれの原子についての平均を、さらに時間平均すると、 $\bar{u} = 0.0177 \times a_0 = 0.18$ [Bohr] という値を得る。この値は rhsi のときの \bar{u} より倍くらい大きくなっている。単位格子が基本単位格子 rhsi から簡易単位格子に拡大されて低振動モードが加わったためと解釈できる。

さらにこの続きとして、温度を変えてみる。今度は、全部で 200 ステップを取り (全時間 0.967 [ps] 相当) その間、初めからステップ 10 まで、初期温度を 300 [K] に保ち、それからステップ 190 まで温度を 50 [K] までリニアに下げる。ステップ 190 以降は温度は 50 [K] のまま保つ。

結果は、図 8.5 の通り、温度の減少に伴い、結晶全エネルギーは振動減衰する様子が解る。また、図 8.5 (b) では、原子の平均変位の時間変化はやはり少しずつ減少しているこ

とが解る。

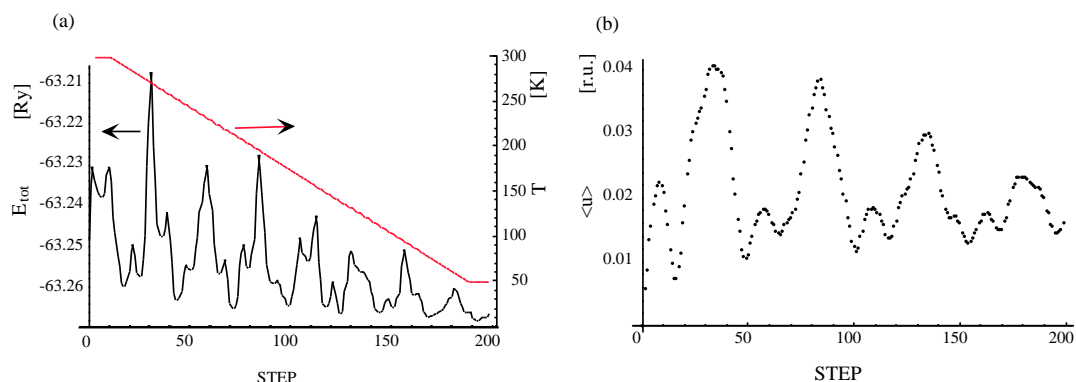


図 8.5: Si_8 の冷却過程。(a) E_{tot} (黒線) と原子の運動エネルギー温度 T (赤線) の変化。時間ステップ 100 [Rysec]。(b) 原子の平均変位 \bar{u} の時間変化。

温度変化という場合、これくらいのゆっくりさでないと本当のシミュレーションにならないだろう。ある温度 T が誤差 ΔT の範囲で意味を持つためには、 T が誤差 ΔT の範囲内にいる間、原子は何度も振動を繰り返す必要がある。

ちなみにこの温度シミュレーションの計算にかかった CPU 時間は、SGI の Origin2000 クラスのワークステーションでだいたい 80 時間であった。それで得られた全シミュレーション時間はわずか 1 [ps] に過ぎない。相転移では長距離波長の揺らぎが重要であることが知られているが、それを再現するにはこの時間スケールでは何桁も短い。Car-Parrinello のおかげで第一原理分子動力学シミュレーションのパフォーマンスが飛躍的に向上したとはいえ、依然として高価な計算には変わらない。

このように、現在の第一原理分子動力学シミュレーションのパフォーマンスからして、現実の相転移をそのまま再現するということはほとんど不可能である。それではもっと早い「温度変化」をさせた場合、それは何の意味も持たないのか？それは目的による。相転移を忠実に再現するという意味では、それは正しくない方法だろう。しかし、単に原子位置の大局的安定位置を探すような場合であれば、温度変化が如何に非現実的であろうと、それは関係ない。ここに第一原理分子動力学シミュレーションが広く使われている理由がある。

終章

このプログラムは、理論には興味ない実験家の人にも使ってもらうことを意図した。しかしながら、そのような意図は果たせたであろうか？実験家の願うことは、原子の種類、位置とかの物理的量だけを入力として、結果がどうなるかを知りたいということで、その過程で計算制御用の人為的パラメータに煩わされるのは望んでいないであろう（その思いは理論家も同じである）。残念ながら現実はそのようなことを各章で見てきたと思う。平面波のカットオフとか、時間の刻みという人為的な操作は、計算上の都合にしか過ぎない。現実の波動関数は無限基底で表されるのであり、時間は連続している。あくまで計算結果はシミュレーションであり、現実をそっくりそのまま再現しているわけではない。²

例としてここに読者に問題を提供する。第8章の分子動力学シミュレーションでは、結晶中の原子位置の時間変化を追跡した。これに対して時間に関するスペクトル分解をすればフォノン振動数、また併せて空間に関するスペクトル分解すればフォノンの分散関係が得られるはずである。このときのフォノンの振動数幅は実際の幅を表すだろうか？フォノンの振動数幅は、フォノンの減衰に関係するが、それはポテンシャルの非調和項の効果として現れる。第一原理計算ではこの効果はきちんと出るので、原理上このシミュレーションでフォノンの振動数幅も再現できるはずだ。しかし...

どうしてもそのギャップを埋めるものが必要で、それが数値計算の原理に対する理解なのだと思う。結果のみを欲する人にはそれは余計な労力かもしれないが、願わくばその余計な労力は、物理的理解とは無関係ではないということだ。

我々が物理現象を理解するときは、多くの場合、単純化されたモデルを使い「ああそうなるのだなあ」と納得する。例えば、格子振動では、原子間はバネで結ばれていると考える。もちろん現実にはそんなバネは存在しないが、しかしそれで良く自然が再現され理解も進む。モデルであれば、我々は初めからそれは自然を単純化したものに過ぎないと認識しているので、現実と混同するという問題は生じない。ところが、第一原理シミュレーションではどこまでが人為的なものであるのかが分かり難い。膨大な数値出力の海の中で、どれが物理的に意味を持つのか戸惑ってしまうが、その判断に本書が少しでも役立てば幸いである。³

終わりに当たって、このパッケージの発展について一言述べなくてはならない。

²作者の大学時代の学生実験で、アナログシミュレーターで微分方程式を解く実験があった。レポートの最後に、シミュレーションと真の解の違いについて書くように指示があった。これは今になって思えば大変重要なことを学んだと思う。最近世の中で、おかしな事件が多発しているのも現実とシミュレーションの区別がつかなくなった事にも一因があるような気がする。

³ここに素晴らしい名言がある。私自身読んで感激した Koonin の計算機物理の本 [43] の初めにピカソの名言が引用されていたが、それをここでもう一度孫引きさせていただく。

「コンピュータの問題は正に答えしか与えないことだ。」

来年のことを言うと鬼が笑うと言うが、プログラム開発に当たってはまさにこの通りである。いくら良いことを言っても、あるいは「原理上このような計算が可能である」と言っても現実にちゃんとプログラムとして実現しないことには話にならない。そしてそのことに膨大な時間がかかることはある程度の規模のプログラム開発経験のある人であれば身を持って感じていることであろう。たった一つのタイプミスによるバグを探すため一ヶ月を費やすということも珍しいことではない。

このような状況であるから、いくら「原理上これはすぐ計算できる」という確信があっても、「ちゃんと動くものを作ってから言うべし」という信念を著者は抱いている。

そういうことをわきまえた上で、敢えて「Osaka2000」の将来発展について述べる。

差し当りは、今すぐできることとして、 d 、 f 電子への拡張はまだ良くテストされていないので、それをする必要がある。

擬ポテンシャルで扱う以上、その物理的描写に限度があるが、それでもスピンを導入することに原理上の困難点はない。

スピンが入れば、行き着くところ電子関連の問題に行き当たる。局所密度近似の限界については数多くの議論がなされており、それを越える試みが数々なされている。電子関連の問題は現在の凝縮系の物理の最重要課題と言っても過言ではない。GW 近似などでどこまでこの問題に迫れるか、など興味は尽きない。

また、近年の擬ポテンシャル法の改良は、ウルトラソフトポテンシャルに見られるように、ある意味で内殻の電荷補正を入れる動きにある。こうして全電子計算との境がだんだん無くなってゆくような気がする。そのような動きを看過するわけにはいかない。

一方、局所密度近似の範囲でも、密度汎関数法の応用は開けてきている。差し当り、線形応答理論と結びついた、系の様々なレスポンス関数の計算に応用されているが、フォノンの分散関係、非調和現象への応用などは是非やってみたいと思っているものである。

さらに、そのような拡張まで行かなくても、今のプログラムのままでさえ、いろいろな応用問題に適用する度に、新たな技術的問題に直面する。そのようなことをこのマニュアルだけで網羅することは不可能である。それでそのような応用に関しては付属するテクニカルレポートシリーズに詳細されるので、参照されたい。

付録 A 結晶の Mathematica によるグラフィックス表示

A.1 グラフィックス座標の取り方

初めに述べたように、「OSAKA2000」のグラフィックス出力はほとんど Mathematica のノートブックに頼っている。その中で電荷密度を描く ChgDnst.nb に関して最低限のことを述べておく。本文では電荷密度分布の 2 次元へのマッピングを強調したが、その 2 次元の切断面の取り方についてである。波動関数の表示も同じだ。どちらにしるこれらの Mathematica ノートブックはバカチョンの使い方を想定してはいない。基本的にはソースコードの解読を前提としている。

ここでは trivial でない例として、単斜晶に属する結晶、 α 型酸素結晶を取り上げる。空間群は C2/m で、底心格子 (C-center) でその結晶形は図 A.1 (a) に示されている。

基本的に、Mathematica では (多分他のグラフィックスプログラムでもそうだと思うが) 密度分布は等方的、かつ直交している立方体メッシュを用いて描かれる。それゆえ斜交軸で表された結晶では、非常に面倒な変換が必要だ。ユーザーはその面倒な変換を知る必要はないが、自分が得たい切断面をどう指定するかを知らねばならない。それはどのような商用のプログラムでもそうで、結局座標系に対する理解なしではどうにもならない。

ChgDnst.nb では、セクション 2. Construct crystal structure までのところで実空間結晶は以下のように記述されている。まずセクション read *.prim の中で、

```
file = OpenRead["name.prim"]
```

と結晶データ *.prim を読み込むので、結晶ファイル名を入力する。

長さの単位は $\{a_0, b_0, c_0\}$ のうち一番大きいもの (\AA 単位で a_{\max}) に取られる。

- 基本単位格子は三つの基本並進ベクトル $\{t_1, t_2, t_3\}$ で表される (ノートブック中で $\{t_1, t_2, t_3\}$)
- 簡易単位格子は三つの基本並進ベクトル $\{t_{c1}, t_{c2}, t_{c3}\}$ で表される (同 $\{tc1, tc2, tc3\}$)
- さらに Mathematica で図示化するためこのセット全てを含むグローバル領域を取る。グローバル領域の基本ベクトル $\{gp1, gp2, gp3\}$ は必然的に直交デカルト座標系である (同 $\{gp1, gp2, gp3\}$)

そのデータにより、ChgDnst.nb は図 A.1 (a) に示されるように、基本単位格子と簡易単位格子を描き、それら両方を含む直方体領域を取る (これを仮にグローバル領域と呼ぼ

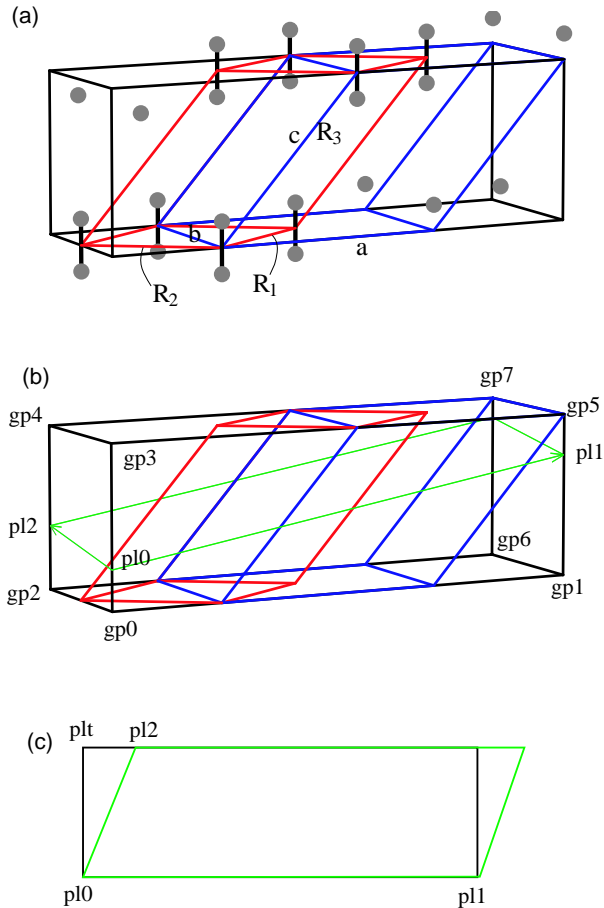


図 A.1: 結晶構造から2次元切断面の切り方。(a) α 型酸素結晶の基本単位格子(赤線)と慣用単位格子(青線)。いずれをも含む直方体を描く。(b) 切断面(緑線)は3点、pl0、pl1、pl2で指定される。(c) 2次元電荷密度分布はpl0、pl1、pltで張られる長方形の中で描かれる。

う)。それらの各頂点の3次元座標は、gp0、...、gp7と名付けられる(同図(b))。以下の座標系はこのグローバル座標系で記述されることになる。

この座標系で

- 基本単位格子の中の基本構造をつくる N_{atom} (natom) 個の原子の位置は $\text{ba}[i]$ で表される。
- 三つの整数の組で表わされる格子は、latticeで

$$\text{lattice}[l_m_n_] := N[l \ t1+m \ t2+n \ t3]$$

で与えられる。しかしながらその内のいくつかは以下のようにシリアル番号で識別されている。

$$\text{lattice}[0]=\text{lattice}[0,0,0];$$

```

lattice[1]=lattice[1,0,0];
lattice[2]=lattice[0,1,0];
lattice[3]=lattice[0,0,1];
lattice[4]=lattice[1,1,0];
lattice[5]=lattice[0,1,1];
lattice[6]=lattice[1,0,1];
lattice[7]=lattice[1,1,1];
lattice[8]=lattice[-1,1,1];
lattice[9]=lattice[1,-1,1];
lattice[10]=lattice[1,1,-1];
lattice[11]=lattice[1,-1,0];
lattice[12]=lattice[1,0,-1];
lattice[13]=lattice[0,1,-1];
lattice[14]=lattice[2,0,0];
lattice[15]=lattice[0,2,0];
lattice[16]=lattice[0,0,2];

```

かつ

```
lattice[-#]=-lattice[#];
```

となる。

- したがって結晶中の原子位置は関数 atomPos により

```
atomPos[l_Integer,j_Integer]:=lattice[l]+ba[j]
```

となる。

以上が主要な変数のリストである。

ところで結晶の図で、今述べた gp0、...、gp7 で作られる領域内にクリッピングしてしまると、今の酸素結晶の場合、O₂ 分子の半分しか見えず、分子構造が見えにくくなる。そのため原子や最近接ボンドに関してはこれより少し広い領域で取る必要があるが、このあたりは物によって違うので、ユーザーがノートブックを解読して制御してもらうということに委ねたい。

その後、セクション read data file で

```
file = OpenRead["pwm_name.rho"]
```

と電荷密度分布のファイルを入力し、データを読み込む。この後、ChgDnst.nb は density statistics の見出しの後、

```
{ro3min, atmin}
{0.234258, {1, 1, 35}}
```

```
{ro3max, atmax}
{247.381, {12, 58, 56}}
```

と電荷密度の最大と最小値を割り出す。

3次元の等電荷密度面は、先程の直方体のグローバル領域でのメッシュ点でデータを内挿しながら描かれる。ユーザーは最後のプロットコマンドの

```
Timing[ grf = ContourPlot3D[ fulst[1,m,n],
  {1,igxmin,igxmax},{m,igymin,igymin},{n,igzmin,igzmax},
  Contours->{80}, PlotPoints->{8,4} ] ]
```

の中で、Contoursのパラメータを、先程の電荷密度の最大と最小値をにらみながら、適当な電荷密度の値を入れるだけで済む。図 5.4 の 3 次元等電荷密度面はこのようにして、gp0、...、gp7 で作られるグローバル領域内で描かれたものである。

次に、2 次元の切断面は、2 つのベクトルで指定する。同図 (b) のように、それは 3 点、pl0、pl1、pl2 で指定される。この点の指定には、点 gp0、...、gp7 を用いた相対座標を用いるよう薦める。すなわち、pl0 を線分 gp0-gp3 の 1/4 の点に取りたいければ、

$$\vec{pl0} = \frac{3\vec{gp0} + \vec{gp3}}{4} \quad (\text{A.1})$$

と取る。

このベクトルで作られる領域は一般には長方形ではないので、同図 (c) のように改めて pl0-pl1 と直角方向に plt が取られる。同図の黒の領域が最終的なプロット領域になる。ChgDnst.nb ではこの領域と、単位格子の混ざる交線や、あればその面上の原子も描く。その例が図 5.5 で示される。

A.2 電荷密度分布

実空間の電荷密度分布データ rod.r はデータ量が必要最小限になるよう、基本単位格子内の空間点でのみ与えられる。そして基本単位格子空間を {na1, na2, na3} で分割しその「格子点」での電荷密度の値で与えられる。その規格化因子は

$$N_{el} = \sum_{i=1}^{NA3} \rho_i \times \frac{1}{NA3} \quad (\text{A.2})$$

と取ってある。N_{el} が基本単位格子中にある電子数である。

電荷密度分布データがノートブックに取り込まれた後、密度データは加工される。考慮すべき点が 2 点ある。

まず、密度分布のプロットなどほとんどの解析のためには、密度は直交座標空間での格子点で与えられないととても解析が複雑になる。一方で元となる基本単位格子は一般には直交座標ではない。たとえ結晶が立方晶で会ったとしても FCC や BCC の場合は基本単位格子はもはや立方体ではない

まず、密度分布のプロットなどほとんどの解析のためには、密度は直交座標空間での格子点で与えられないととても解析が複雑になる。一方で元となる基本単位格子は一般には直交座標ではない。たとえ結晶が立方晶で会ったとしても FCC や BCC の場合は基本単位格子はもはや立方体ではない

もう一つが、密度データは基本単位格子空間の中だけで展開されているので、3 次元空間の一般点の密度を求めるとき、その点を基本単位格子空間の等価な点に引き戻さなければならぬ。

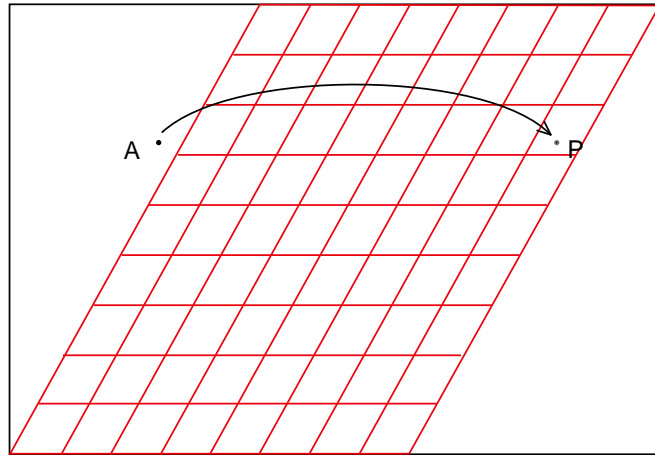


図 A.2: 電荷密度分布における座標変換。基本単位格子は赤線で示されている。電荷密度データはその「格子点」で与えられている。黒枠はグローバル領域。直交座標で与えられた点 A は基本単位格子空間の等価な点 P に引き戻される。

以上の要請から、3次元直交座標系から、電荷密度分布が与えられている基本単位格子空間との座標変換が必須である。その関係が図 A.2 で示されている。

その座標変換をし、ある点の密度を計算するものがノートブックの関数

`fd[l_,m_,n_]`

である。により参照する。この関数は3次元空間の任意の一点の電荷密度を与える。電荷密度を求めたい空間点 A は直交座標で $\{l, m, n\}$ で与える。その座標を図 A.2 に従い、 $\{na1, na2, na3\}$ 分割の基本単位格子空間内の等価な点 P に引き戻す。この P 点は一般には基本単位格子空間の「格子点」にはなっていないので、その電荷密度の値はその近くの「格子点」上の電荷密度から内挿する。その内挿値を `fd` は返す。

3次元や2次元切断面での電荷密度分布は全てこの `fd` を通じて描かれている。

謝辞

本プログラムを開発するに当たり、先駆者の貢献については序論の歴史で述べた。ここではそれ以外の人への貢献について述べたい。プログラム開発には、実践でのテストが必要不可欠だ。その意味では、特に、開発途上のプログラムで仕事をしてくれた二人の学生、藤田洋土、山崎勇治君には試験台になってもらって、ここにその貢献に対して感謝したい。

References

- [1] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964)
- [2] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965)
- [3] J. Callaway and N. H. March, *Solid State Physics* **38**, (1984) p. 135
- [4] R. O. Jones and O. Gunnarsson, Rev. Mod. Phys. **61**, 689 (1989)
- [5] *Adv. in Quantum Chemistry* **21**, S. B. Trickey ed., (Academic, San Diego, 1989)
- [6] D. M. Ceperley and B. Alder, Phys. Rev. Lett. **45**, 566 (1980)
- [7] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981)
- [8] V. Heine, *Solid State Physics* **24**, (1970) p. 1, and all the volume.
- [9] D. R. Hamann, M. Schlüter, and C. Chiang, Phys. Rev. Lett. **43**, 1494 (1979)
- [10] N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991)
- [11] L. Kleinman and D. M. Bylander, Phys. Rev. Lett. **48**, 1425 (1982)
- [12] G. B. Bachelet, D. R. Hamann, and M. Schlüter, Phys. Rev. B **26**, 4199 (1982)
- [13] N. W. Ashcroft and N. D. Mermin, "Solid State Physics", (Holt-Saunders, Philadelphia, 1976)
- [14] J. Ihm, A. Zunger, and M. L. Cohen, J. Phys. C: Solid State Phys. **12**, 4409 (1979)
- [15] M. Born and K. Huang, "Dynamical Theory of Crystal Lattices", (Oxford University Press, Oxford, 1954)
- [16] H. J. Monkhorst and J. D. Pack, Phys. Rev. B **13**, 5188 (1976)
- [17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. R. Flannery, "Numeric Recipes", 2nd ed. (Cambridge, Cambridge, 1992)
- [18] M. P. Teter, M. C. Payne, and D. C. Allan, Phys. Rev. B **40**, 12255 (1989)
- [19] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, Rev. Mod. Phys. **64**, 1045 (1992)
- [20] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985)

- [21] R. P. Feynman, Phys. Rev. **56**, 340 (1939)
- [22] O. H. Nielsen and R. M. Martin, Phys. Rev. B **32**, 3780 (1985)
- [23] O. H. Nielsen and R. M. Martin, Phys. Rev. B **32**, 3792 (1985)
- [24] 柳瀬章「空間群のプログラム」裳華房 (1995)
- [25] T. Hahn, ed. "International Tables for Crystallography", Vol. A (Kluwer Academic Pub., Dordrecht, 1996)
- [26] G. Burns and A. M. Glazer, "Space Groups for Solid State Scientists", 2nd. ed., (Academic, San Diego, 1990)
- [27] M. T. Yin and M. L. Cohen, Phys. Rev. B **26**, 3259 (1982)
- [28] J. Ihm and M. L. Cohen, Phys. Rev. B **21**, 1527 (1980)
- [29] J. Stoer and R. Bulirsch, "Introduction to Numerical Analysis", 2nd ed. (Springer, New York, 1992)
- [30] D. J. Chadi and M. L. Cohen, Phys. Rev. B **8**, 5747 (1973)
- [31] C. H. Park, B.-H. Cheong, K. H. Lee, and K. J. Chang, Phys. Rev. B **49**, 4485 (1994)
- [32] J. Callaway, "Quantum Theory of the Solid State", 2nd. ed., (Academic, San Diego, 1991).
- [33] C. -L. Fu and K. -M. Ho, Phys. Rev. B **28**, 5480 (1983)
- [34] M. R. Pederson and K. A. Jackson, Phys. Rev. B **43**, 7312 (1991)
- [35] M. T. Yin, *Proc. 17th Int. Conf. on the Physics of Semiconductors*, eds. J. D. Chadi and W. A. Harrison (Springer, NY, 1984) p. 927
- [36] O. Jepsen and O. K. Anderson, Solid State Commun. **9**, 1763 (1971)
- [37] C. Herring, Phys. Rev. **52**, 365 (1937)
- [38] G. F. Koster, *Solid State Physics* **5**, (1957) p. 173. 指標表はとかくミスプリントがありがちだ。脈絡のない数字の羅列だから不思議はない。実際この Koster の論文にもそれがある。この Koster のも含めて、主要な指標表のミスプリントの正誤表は、J. L. Warren, Rev. Mod. Phys. **40**, 38 (1968) の付録にある。
- [39] O. V. Kovalev, "Representations of the Crystallographic Space Groups", 2nd ed., translated to English (Gordon and Breach Scientific Pub., Yverdon, 1993)
- [40] C. J. Bradley and A. P. Cracknell, "The Mathematical Theory of Symmetry in Solids", (Clarendon Press, Oxford, 1972)

- [41] L. Verlet, Phys. Rev. **159**, 98 (1967)
- [42] D. Frenkel and B. Smit, "Understanding Molecular Simulation", (Academic, San Diego, 1996)
- [43] S. E. Koonin and D. C. Meredith, "Computational Physics", (Addison Wesley, Massachusetts, 1990)