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

^r Osaka2002_*nano* J

Ⅰ 「基礎編」

第1.6版

大阪大学・産業科学研究所

白井光雲

2 November 2004

History

Rev. 0.7 04 March 2003

Rev. $0.9\ 18$ December 2003

Rev. 1.0 04 February 2004

Rev. 1.1 02 November 2004

Rev. 1.5 29 December 2005

Rev. 1.6 06 April 2006

Rev. 1.7 11 May 2007

目 次

序論		4
第1章	理論	7
1.1	密度汎関数法	$\overline{7}$
1.2	局所密度近似	9
1.3	擬ポテンシャル近似	9
1.4	平面波展開	11
1.5	平面波のカットオフ	13
1.6	k 空間上の総和	14
1.7	Kohn-Sham 汎関数における共役勾配最小法	15
1.8	高速フーリエ変換................................	19
1.9	Hellmann-Feynman カとストレス	20
第2章	ロードマップ	22
2.1	概略	22
2.2		23
	2.2.1 機種ごとの対応	25
2.3	入力パラメータ	25
	2.3.1 オプション	26
箆3音	百子擬ポテンシャル	27
31		27
3.2		30
0.2		00
第4章	結晶の作成	33
4.1	結晶データ入力	33
4.2	cryst の実行	36
4.3	制御パラメータ	38
4.4	結晶のグラフィックス表示	38
第5章	電子の基底状態計算(I)	40
5.1	inip	40
	5.1.1 k サンプリング法	41
	5.1.2 平面波カットオフ半径	44
5.2	pwm の動作	46
	5.2.1 セルフコンシステント 計算	46

		5.2.2 解釈	. 48
		5.2.3 再計算	. 52
		5.2.4 電荷密度の表示	. 54
		5.2.5 収束がうまくいかない場合	. 56
		5.2.6 関係するオプション	. 57
筜	6 音	雪子の其広(北能計質 / II)	50
ন	0 早 6 1		50
	0.1		. 59
		0.1.1 前昇の午備	. 59
		0.1.2 約未の読の月	. 00
		0.1.3 月の設定に関して	. 01
	6.0	0.1.4 以米に送りる 	. 62
	0.2	格丁疋奴の取週11	. 64
			. 64
			. 65
			. 66
		6.2.4 圧力依存性の取り方	. 67
		6.2.5 関係するオフション	. 68
第	7章	電子構造スペクトル	69
	7.1	DOS 構造	. 69
		7.1.1 準備	. 69
		7.1.2 DOS 構造の結果	. 71
		7.1.3 表示	. 73
		7.1.4 フェルミ面	. 75
	7.2	バンド構造	. 75
		7.2.1 準備	. 76
		7.2.2 バンド構造の結果	. 77
	7.3	波動関数のプロット	. 83
付	録A	結晶の Mathematica によるグラフィックス表示	84
	A.1	グラフィックス座標の取り方............................	. 84
	A.2	電荷密度分布	. 87
謝	锌		89

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 れてきていている。化学の分野ではもうそのようになっているような気がするが、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障害であろう。

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がこれが かなり初心者に負担になっていた。これを間違うと、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パリわか 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ま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に変わる。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タレーションマニュアル 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。従って更なる応用を目指す読 者はそれらを参照していただく。

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。その意味でコードに関し ても著者に責任がある。

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ので、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 ならない。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 まで御願いする。

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 他の教科書で勉強する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、

$$H = -\sum_{i} \nabla^{2} + \frac{1}{2} \sum_{i \neq j} \frac{e^{2}}{|\mathbf{r}_{i} - \mathbf{r}_{j}|} + \sum_{i} V_{\text{ion}}(\mathbf{r}_{i}) + E_{\text{ion}}(\{\mathbf{R}_{n}\})$$
(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、

$$H\Psi_0(\{\mathbf{r}_i\}) = E_0\Psi_0(\{\mathbf{r}_i\})$$
(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ない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 Xα 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ものパーツか 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、Rydburg 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。

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

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。

$$\rho(\mathbf{r}) = \sum_{i} f_{i} |\Psi_{i}(\mathbf{r})|^{2}$$
(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。

$$H_{\rm KS}\Psi_i = \epsilon_i \Psi_i \tag{1.5}$$

ここで、H_{KS}はKohn-Shamのハミルトニアンであり、

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

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

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

と表され、V_{xc} は交換相関ポテンシャルである。

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

 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。

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

交換相関ポテンシャル $V_{\rm xc}(\mathbf{r})$ は、

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

そして、 $\epsilon_{\rm xc}({\bf r})$ は

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

で、 ϵ_{xc}^{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のには適切ではない。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、

$$\hat{V}_{NL} = \sum_{l} |lm\rangle V_l \langle lm| \tag{1.12}$$

ここで |*lm*〉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、

$$V_{\rm ion} = V_{\rm L} + \sum_{lm} \frac{\left| \Phi_{lm}^0 \Delta V_l \right\rangle \langle \Delta V_l \Phi_{lm}^0 |}{\langle \Phi_{lm}^0 | \Delta V_l | \Phi_{lm}^0 \rangle}$$
(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、

$$\Delta V_l = V_{l,\rm NL} - V_{\rm L} \tag{1.14}$$

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

マクロな結晶では~10²³ 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、

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

で与えられる。n₁,n₂,n₃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

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

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。 G_1 は以下のようになる。

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

 G_2, 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。

$$\mathbf{G} = n_1 \mathbf{G}_1 + n_2 \mathbf{G}_2 + n_3 \mathbf{G}_3 \tag{1.18}$$

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、

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

となる。AG は逆格子ベクトル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、

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

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、

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

従って平面波法はそのままで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

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

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 カ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。

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。

式(1.22)の最後の項は、イオンー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

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

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

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

で与えられる。プライムのついてい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

$$\left|\mathbf{k} + \mathbf{G}\right|^2 < E_{\rm cut} \tag{1.24}$$

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フ半径は

$$E_{\rm cut} = k_{\rm c}^2 \tag{1.25}$$

で与えられる。

カットオフ半径 kc 内の平面波の数 N_{DW} は

$$N_{\rm pw} = \frac{\frac{4\pi}{3}k_c^3}{\frac{(2\pi)^3}{\Omega_c}}$$
(1.26)

と見積もれる。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。



図 1.2: a) 点 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。

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 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、

$$Q = \frac{\Omega_{\rm c}}{(2\pi)^3} \int_{\rm BZ} Q_{\bf k} d^3 {\bf k}$$
(1.27)

ここで $Q_{\mathbf{k}}$ は、

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

と表される。例えば、 \hat{Q} が Kohn-Sham ハミルトニアンである場合、 Q_{kn} 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 N₁, N₂, N₃ 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。

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

$$u_{ip} = \frac{2p - N_i - 1}{2N_i} \tag{1.30}$$

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。

$$Q = \frac{1}{N_1 N_2 N_3} \sum_{rst} Q_{\mathbf{k}_{rst}}$$
(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 点が対称ならば、Kohn-Sham 方程式は対称な点のうちの一点での 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。

$$Q_{\mathbf{k}} = \sum_{\mathbf{R}} B_{\mathbf{R}} e^{i\mathbf{k}\cdot\mathbf{R}} \tag{1.32}$$

ここで $B_{\mathbf{R}}$ は展開係数である。ゾーン全体にわたる $Q_{\mathbf{k}n}$ 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、

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

 $\phi_{\mathbf{R}}$ は

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

と書け、サンプリング \mathbf{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、

$$\sum_{\mathbf{R}}^{i\text{th shell}} \phi_{\mathbf{R}} \tag{1.33b}$$

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、系の基底状態を求めることは Kohn-Sham のエネルギー汎関数 $E_{\text{KS}}(\{c_{\mathbf{k}+\mathbf{G},n}\})$ 関数を占有された波動関数の平面波展開係数 $\{c_{\mathbf{k}+\mathbf{G},n}\}$ について最小化さ 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

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

と線形近似でき、かつ 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

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

であるので、 $f(\mathbf{x})$ が極小値をとるときの \mathbf{x} は線形方程式

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \tag{1.36}$$

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一次元最小化を行う。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 \mathbf{h}_i 上で最小点 \mathbf{x}_i が見つかったとする。次にこの点で勾配 \mathbf{g}_i を求める。 \mathbf{x}_i を始点として、新しい i+1番目のステップ探索方向 \mathbf{h}_{i+1} は

$$\mathbf{h}_{i+1} = \mathbf{g}_i + \gamma_i \mathbf{h}_i \tag{1.37}$$

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

$$\gamma_i = \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i} \tag{1.38}$$

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 · 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ーの節約にもなる。

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で、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化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。

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 { $c_{\mathbf{k}+\mathbf{G},n}$ } = \mathbf{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 \mathbf{C}_n^m に関する期待値は

$$\lambda_n^m = \mathbf{C}_n^{m*} \cdot \mathbf{H} \cdot \mathbf{C}_n^m \tag{1.39}$$

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

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

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

この残留誤差ベクトル 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、

$$\mathbf{R}^{\prime m} = \mathbf{R}^m - \sum_{r \neq n} \left(\mathbf{C}_r \cdot \mathbf{R}^m \right) \mathbf{C}_r \tag{1.41}$$



図 1.4: 直接最小化法のフローダイアグラム。

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

$$\mathbf{R}^{\prime\prime m} = \mathbf{R}^{\prime m} \cdot \mathbf{K} \tag{1.42}$$

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

4. 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。

$$\mathbf{G}^{m} = \mathbf{R}^{\prime\prime m} - \sum_{r \neq n} \left(\mathbf{C}_{r} \cdot \mathbf{R}^{\prime\prime m} \right) \mathbf{C}_{r} - \left(\mathbf{C}_{n}^{m} \cdot \mathbf{R}^{\prime\prime m} \right) \mathbf{C}_{n}^{m}$$
(1.43)

5. この \mathbf{G}^m を使って、共役勾配を求める。つまり

$$\mathbf{F}^m = \mathbf{G}^m - \gamma^m \mathbf{F}^{m-1} \tag{1.44}$$

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 $\gamma^0 = 0$ 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

$$\mathbf{C}_{n}^{m+1} = \mathbf{C}_{n}^{m} \cos \theta + \mathbf{D}^{m} \sin \theta \tag{1.45}$$

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、

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

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、

$$\left. \frac{\partial E_{\rm KS}}{\partial \theta} \right|_{\theta=0} = 2f_n {\rm Re}(\mathbf{D}^{m*} \cdot \mathbf{H} \cdot \mathbf{C}_n^m)$$
(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、 $E_{\rm KS}(\theta)$ を0でない他の点 θ_1 で評価するか、あるいは $\theta = 0$ での $E_{\rm KS}(\theta)$ 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 $A = (1/4)\partial^2 E_{\rm KS}/\partial\theta^2$ で与えられる。これらより、式(1.46)を最小にする $\theta_{\rm min}$ は

$$\theta_{\min} = -\frac{1}{2} \tan^{-1} \left[-\frac{\frac{\partial E_{\text{KS}}}{\partial \theta}\Big|_{\theta=0}}{\frac{1}{2} \frac{\partial^2 E_{\text{KS}}}{\partial \theta^2}\Big|_{\theta=0}} \right]$$
(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 $N_{\rm pw}$ 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 $\mathcal{O}(N_{\rm pw})$ 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 $\mathcal{O}(N_{pw}^2)$ 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。

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

式(1.20、1.21)より、G'についての足し合わせは

$$u_{\mathbf{k}n}(\mathbf{r}) = \sum_{\mathbf{G}'} e^{i\mathbf{G}' \cdot \mathbf{r}} c_{\mathbf{k}+\mathbf{G}',n}$$
(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 $V_{\mathrm{L}}(\mathbf{r})$ 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 $\mathcal{O}(N_{pw})$ の演算で済み、1と3はFFTを用いると $\mathcal{O}(N_{pw}\log(N_{pw}))$ の演算で済み、結果として全体でも $\mathcal{O}(N_{pw}\log(N_{pw}))$ 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辺の長さが $4k_c$ 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

$$N_{\rm FFT} \cong 16N_{\rm pw} \tag{1.51}$$

となる。

1.9 Hellmann-Feynman 力とストレス

Hellmann-Feynman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。

$$\mathbf{F}_{I} = -\frac{dE_{\text{tot}}}{d\mathbf{R}_{I}} = -\frac{d}{d\mathbf{R}_{I}} \langle \Psi | \hat{H} | \Psi \rangle \tag{1.52}$$

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-Feynmanの定理によると、式(1.52)は、

$$\mathbf{F}_{I} = -\langle \Psi | \frac{\partial V(\{\mathbf{R}\})}{\partial \mathbf{R}_{I}} | \Psi \rangle \tag{1.53}$$

と変形できる(引用としては当然 Hellmann と Feynman 両方をするべきであるが、作者は Feynman[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。

カ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つのディレクトリができ、さらにユーザーが/bin ディレクトリ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 が収めされている。 全てのソースコードは Makefile とともに、ディレクトリー/fsrcs に置かれている。それ を/bin ディレクトリにコピーしこのディレクトリ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ーディレクトリ、 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 fsrcs 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 のインストレー ションマニュアル」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 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ーつで済む。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 pwbcd 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。

自分の機種に会った 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'

の部分を自分の環境のポテンシャルデーター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 パイラーによって対処方法が異なるので「Osaka2002 のインストレーションマニュアル」 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 のインストレーショ ンマニュアル」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。

2.3 入力パラメータ

Osaka2kを構成するどのプログラムも、それぞれの専用入力パラメータが必要である。 それらは一般に*.paraの名前のファイルで用意される。それらの書式はそれらが現れる 順に説明していくが、ここでどの場合にも共通の事項について述べる。

これまでのプログラム開発の経験で明らかになったことは、プログラムの仕様が拡大されるにつれ、それらの入力パラメータが変わってしまい、違ったバージョンのプログラムを使うと動かなくなるということである。またそのような追加されたパラメータは多くの場合上級者のためのもので平均的なユーザーはほとんど使わないことが多い状況がある。そこで*.paraに常時書き入れるパラメータは平均的ユーザーが良く使うものを選び、後は必要に応じてオプション入力にするという方法に変えた。

2.3.1 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. 值変数

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 を参照してもらう。あるいはその入力部分をまとめた opts_list が参考に なる。

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。

itype	ititle				
i					
name at	icorr + ispp				
znuc	zsh	rsh	rmax	aa	bb
n_{core}	n_{val}				
n	l	zo_{\downarrow}	zo_{\uparrow}	evd	
r_{cs}	r_{cp}	r_{cd}	r_{cf}	cfac	rcfac

Table 3.1: atom.dat のフォーマッ	┝
----------------------------	---

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 interporation 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

 \sqcup 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rcfac)。

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るのではなく、 p_x 、 p_y 、 p_z それぞれを平等に 1/3 づ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)			
2.1	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

 $\mathbf{tm2}$ 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。

第五行目は、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、スピンダウン、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、スピンダ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とき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 を適当なプロッティングソフトで読めばよ い。

ファイル atom.out では

radial grid parameters

n

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、

= 2	1 = 0 = .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 $1.1 \sim 1.6$ 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 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 に書き込る。

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 などを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ことが望ましい。

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) はそのままで 規格化されている。すなわち

$$\int_{0}^{\infty} |u(r)|^2 dr = 1$$
 (3.1)

であり、∫···rdr 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。今の場合は 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替えす。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。それに関しては別巻で取り扱うことになる。

改定 1/2005 擬ポテンシャルを作るプログラムは、この間にも発展しており、atomは バージョン5へと改定された。一番大きな変更点はGGAが取り入れら れたことである。この取り扱いが一般的になるにつれ、このポテンシャ ルを使うことの要請が大きくなったので、今回のバージョンでは組み入 れている。 このバージョンのものはディレクトリー/patom5 に収められている。オリジナルに比べ てファイルの入出力の部分だけを Osaka2k に合うように直してある。使い方は GGA に関 する入力の部分の他は、これまで述べたことと変わりない。したがって GGA を使おうと しないならば、旧バージョンで十分である。



図 3.1: 改良されたポテンシャルディレクトリ

GGA は別としても、Osaka2k も取り扱うポテンシャルの型が、内殻補正、スピン偏極、 相対論効果などと増えるにつれ、ポテンシャルディレクトリの整理が必要となった。従来 はポテンシャルの種類としては、スピン、内殻補正、相対論補正の3つを区別してきた。 それに応じたポテンシャルデータはspn、nlc、relと三通り用意してきたが、しかしそれ らの組み合わせも出てくるし(例えばスピンと内殻補正)、さらにGGA も取り入れると その組み合わせの数は途端に膨れ上がる。また、取り扱う原子が複数の時、どれが優先さ れるかの指針も明確にしないと混乱に陥る。

そこで今回の改定で、図 2.2 で示されるポテンシャルディレクトリは図 3.1 のように階 層構造を導入し再構築した。それぞれのディレクトリには表 3.2 のような役割を持たせて いる。

dir	potential type
nom	normal potential
nlc	nonlinear core correction
spn	spin polarization
SCC	spin polarization with nonlinear core correction
rel	relativistic effects

Table 3.2: Potential directories.

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、およびその中で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。

ー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、

$$\mathbf{a} = a\hat{x}$$

$$\mathbf{b} = b(\cos\gamma\hat{x} + \sin\gamma\hat{y})$$

$$\mathbf{c} = c(\cos\beta\hat{x} + r\hat{y} + \sqrt{1 - \cos^2\beta - r^2}\hat{z})$$
(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方向に持ってくるように座 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	Х	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 QUALIFIIER 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のも 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

 $N_{\rm at}$: 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 では対称性で結ばれなければ別種として数えるということである。例としてグラファイト を考えると

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

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山 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

値はポテンシャルデータから読まれる。

UNIQUE_b,CELL_1 or 2 or 3

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は重要なので、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
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、そうな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。

cryst の行うこと、制限についての注意

ここで改めて cryst の行うことの精神について述べておく。実際の多様な結晶を取り扱うときいろいろ制限が出てくるからだ。結晶のデーターとしてまず格子定数が必要であるが、この理解には困難点はない。しかし結晶の詳細を記述するときこれ以外に結晶の対称性、空間群が必要で、その理解は大変難しい。一応の基礎知識をマスターしてさえも、実際のところある結晶が与えられたとき、それがどの空間群であるかを決定することは相当の経験が必要である。

著者が意図したことは、そのような結晶の対称性の知識がなくとも計算ができることで ある。そうでないと、多種多様の結晶をいろいろ計算することができないことになるから である。しかしそれは目指すべくして容易に到達しえないものである。そこでユーザーに は結晶のデーターとして、格子定数以外に空間群を指定することを要求した。空間群が何 のことか分からないユーザーには酷かも知れないが、その意味するところが分からなく ともともかくも結晶データを調べれば(あるいは誰かに聞けば)分かるので、それは入力 データとして入れるようにした。一方、原子位置の情報に関しては相対座標のみを入力条 件とした。本来は各原子のワイコフ位置の情報も要るのであるが、これはかなり専門的な 知識が必要となる。そしてそれはITCが手元になくてはなかなか意味するところが分から ない。そこでそのワイコフ位置の決定に関しては cryst が面倒を見ようということだ。こ れは一見、簡単なように見え、実際にやってみると大変難しい。一つ一つの例を見ると、 簡単に見えるが、230 の空間群全てに通用する汎用なアルゴリズムが大変難しい。

この点で、ユーザーにあらかじめ告知しておくべきは、cryst は与えられた原子位置か らワイコフ位置を完全に導くほど賢くないということだ。著者が厚かった経験内である程 度正しく割り振ることができる程度である。新しい結晶をやればやるほど新しいエラーが 見つかるだろう。与えられた結晶の空間群と、割り出したワイコフ位置に整合性がない場 合、cryst はそこでエラーメッセージを残しストップする。人間のアシストが必要となる。 cryst が分からないという答えを出したときはユーザーが代ってワイコフ位置を決めてや らねばならない。このあたりの詳細は Osaka2k の supplement の INPUT GUIDE「cryst に関する覚書」を参照して戴きたい。

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 Fri Jan 5 17:56:20 2001 date: DIMENSION 3 LATTICE PARAMETERS (A,B,C,CA,CB,CC) in a.u. 10.6829045 10.6829045 10.6829045 0.0000000 0.000000 0.0000000 Space group 216 Td2 F-43m ORIGIN_1 IL NG NC 2 24 ORI 1 The conventional vectors 10.6829045 0.0000000 0.000000 0.0000000 10.6829045 0.000000 0.0000000 0.000000 10.6829045 The primitive vectors 5.3414522 0.0000000 5.3414522 5.3414522 0.000000 5.3414522 0.000000 5.3414522 5.3414522 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
                 31
                          3
         ga
                 33
                          5
  2
         as
KIND OF ATOMS
   2
Wycoff Positions
  ATM (
              x,
                              z)
                                      Nos
                                              Wycf
                                                            Code
                      y,
  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			
POSITION	S RELATIVE	TO A UNIT	CONVE	ENTIONAL CELL	SPEC	CIES	SYM(IG)
1	0.000000	0.000	0000	0.000000	1	ga	1
2	0.2500000	0.2500	0000	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読するが、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読できない。³解読に失敗すると 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。たいていの 目的には、TSPARAMとxtl.incは初期値のままで問題ないと思う。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) PARAMETER (MAXNPW=4854)

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

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

³この解読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。

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ー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まるが、だいたいノートブックのタイトルに 沿いながら、ユーザー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 CrystAnal.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。

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 ファイルの 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。

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。これについては次節で述べる。

入力ファイル inip.para は以下のようになる。

```
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 kc を、逆格子ベクトルの中で最小の大きさ gmin を単位に入力。
- k サンプリング点の与え方
 1 で自動的に計算する。0 は手入力。
- k サンプリング分割数 (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、相対論的効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サンプリング法

k サンプリング点の決め方には以下の方法がある。

(i) 自動計算(一様サンプリング=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リ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リ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-Park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-Park 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-Park 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 空け、

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。次の行で与えれる初めの3つの数字は基本ベクトルg1、g2、g3 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、これが間違う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
        0 4
1
    0
          0
 0
    1
             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]。

どちらの方法にしろ、その k サンプリングの質がどれくらい良いものかを、式(1.33b) 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 (-1.00000, 0.00000) 8 1.00000, 15 (0.00000) 23 (-1.00000,0.00000) 26 (-1.00000,0.00000) 28 (1.00000, 0.00000) 32 (-1.00000, 0.00000)1.00000, 0.00000) 36 (38 1.00000, 0.00000) (

とリストされている。ここでは式(??)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。 その他のオプション k 点サンプリングに関しては、これまで述べられたもの以外に、時々 使うものがいくつかある。それらを列挙すると、

1. k 点の原点の選択

Monkhorst-Pack 法ではk 点サンプリングの原点はデフォルトでは1点サンプリン グ以外は、k = 0を取らない。しかし時々k = 0を取りたいときがある。そのときは

KPorg=1

とする。

2. ダブル k 点の選択

これは主に、電子数が奇数の時のためものである。電子数が奇数の時このオプションを設定してやると、同じ k 点に対して、電子数を1個増やしたときと1個減らした場合の計算を実行する。

double_k ON

とする。

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、

$$k_{\rm c} = {\rm AMAX} * g_{\rm min} \tag{5.1}$$

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 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
           1 -1 -1/ 4
Name: XY
                           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。ここにユーザーが制御す 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。はじめに述べたよ うに、それらの付加的なパラメータは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 (maxIterOwfn)
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 (maxIterOatom)
4
Resume cell relaxation
0
max number of iterations for cell (maxIterOcell)
2
```

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 \mathbf{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 ーつ一つ繰り返す。そしてそれを全ての \mathbf{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 maxIterOwfn であろう。npathOwfn 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 maxIterOwfn 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(imoniOctrl) 電荷密度(irhout) 波動関数(iwfout) がある。デバグ時以外はimoniOctrl=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。 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。計算の長くかか るものは、バックグラウンドで流すべきである。

% 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 カ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もらう。

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 べてみて欲しい。

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

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、

it	er	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	0.00009096
CG	process	is stopped bec	ause increase in H	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: 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 npath 回繰り返されるが、繰り返しが進むと npath 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。

が表示される。 $E_{\rm el}$ 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、 $|\Delta E| < {\rm etol}$ 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 $E_{\rm el}$ にイオンの Ewald エネルギー $E_{\rm ewald}$ を加えて、全結晶エネルギー $E_{\rm tot}$ 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ことは危険である。

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/Bohr ³)		
	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、

$$\sum_{i,\mathbf{G}} |c_{\mathbf{k}_i+\mathbf{G}}|^2 (\mathbf{k}_i+\mathbf{G})_{\alpha} (\mathbf{k}_i+\mathbf{G})_{\beta}$$
(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考えるのは誤りである。つ まり両者は数値的に等しくないのである。Hellmann-Faynman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	-FE	YNMAN	FORCE	(RY/BOHR)		
		ATOM		Х	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、それを見つけれ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、まず何をもって誤差というかがをはっきりさせなければならない。入力側にでは(i)入力データの誤差、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、イオンー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、

$$\frac{1}{6}\frac{f_0''}{f_0}h^2\tag{5.3}$$

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みる。表 5.1 にその数値実験結果が出ている。

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、まぁ まぁの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*_{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。

$A_{\rm max}$	$E_{\rm cut}$	$N_{\rm pw}$	NADIM	$E_{\rm el}$	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

Table 5.1: Si における平面波数と E_{el} の関係

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、

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

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。具体的には、平面波数を固定し、原子配置 x をわずかに違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返して再度計算する。出力ファイルpwm_*.wfnをinip_*.wfnとリネームし、pwm.paraの中の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



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

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 $E_{coh}(A)$ は

$$E_{\rm coh}(A) = E(A^{\rm (gas)}) - E(A^{\rm (sol)})$$

$$(5.5)$$

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

$$E_{\rm form}(A_{\rm m}B_{\rm n}) = mE(A^{\rm (sol)}) + nE(B^{\rm (sol)}) - E(A_{\rm m}B_{\rm n})$$
(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が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。



図 5.4: グラファイトの等電荷密度面

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ァイトの計算例で図 5.4 に示す。

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



Max= 0.334983 at {38, 8}

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

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るかわり、文字通り 全体の一面の情報しか与えまないので、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。というと、「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

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~10²³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てはいけない。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。

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、少ししか知らない人が「素人」と呼ばれるのであろう。

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一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。

この場合取るべき対処として、第5.1.1節で説明されたが、k 点を倍にとり、一つは電子数は一つ増やし、もう一つは1つ減らすというダブルk サンプリング法がある。どちらの場合も電子数は偶数になり、奇数の問題はなくなる。これは絶縁体の場合有効である。

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 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してはいけない。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。
- 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、今の場合では縮退である。

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

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。

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一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、一次元最小化は式(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、20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。

5.2.6 関係するオプション

- diagonalize OFF
 デフォルトでは価電子帯のつくる部分空間の中で行列を対角化するが、それをしない。
- allow_posE ON
 エネルギー最小化過程の途中でエネルギーが増加する場合でも繰り返しを続ける。
- $ran_wfn =$

1

初期波動関数の型。デフォルトの1で乱数から作られる。特定の平面波だけから作 りたいときなどはこれを0とセットする。

- vary_occ ON 占有数を可変とする。(デフォルトでは固定)
- energ_deg_Ef=
 1.0E-4
 縮退かどうかの基準値。
- nel_add=
 1
 電子を与えられた数だけ加える(減じる)。整数値。
- $rnel_add =$

0.9

電子を与えられた数だけ加える(減じる)。実数値。

• $nband_extra =$

1

バンド数をデフォルトより与えられた数だけ増やす。

• $mix_occ=$

1.0D-1 占有数を可変としたとき、更新のときの古い値と、新しい値を混ぜ合わせる度合い。 (デフォルト値 1.)

- fermi_broadening ON
 電子の占有数をフェルミ分布関数により与える。
- T_elec_start0fermi=
 0.1
 フェルミブロードニングを掛けるときの初期電子温度。
- T_elec_end0fermi=
 0.1
 フェルミブロードニングを掛けるときの最終電子温度。

このように多くのものは電子占有数の変化に関係するパラメータである。これらの使い 方の実例は Technical Report シリーズにある。関係するものは、

- No. 14 「電子数の制御 エネルギーギャップへの応用 」
- No. 17 「半導体中の不純物準位」
- No. 37 「奇数個電子数の場合の電子占有数の扱い」

などである。

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 $E_{tot}({\mathbf{R}_i})$ 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 (maxIterOwfn) 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 (maxIterOatom)
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 max-Iter0atom 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 \max Iter0atom を超えるか、残留原子間力がftol (R_y/a_B) 以下になるかで終わる。

ー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。引き継ぐためには pwm_*.var のファイルを inip_*.var とリネー ムする。波動関数のデータも前の出力 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 適化の計算をしてみる。

POSITIONS	RELATIVE TO) A UNIT CONVEN	TIONAL CELL	TYPE		SYM(IG)
1 (0.0200000	0.0000000	0.000000	1	si	1
2 (0.000000	0.5000000	0.5000000	1	si	1
3 (0.5000000	0.0000000	0.4900000	1	si	1
4 (0.5200000	0.4800000	0.000000	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 conventi	onal 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に小さくできる。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。51 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

$$\Delta f = \sqrt{2k\Delta E} \tag{6.1}$$

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

$$\Delta x = \frac{\Delta f}{k} \tag{6.2}$$

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 $k_r \simeq 0.3 [Ry/Bohr^2]$ くらいだ。それで残留 Hellnamm-Feynman 力が $1 \times 10^{-3} [Ry/Bohr]$ 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)がわかっているので、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

$$x_{\min} = -\frac{b}{2a} \tag{6.3}$$

と決定される。ここに、*a* と*b* はそれぞれ

$$b = -f$$

$$a = \frac{1}{x_1} \left(\frac{E_1 - E_0}{x_1} - b \right)$$
(6.4)

で与えらる。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、 $x_0 \ge x = x_1$ 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みえるが、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 x₀ ともう一つの試行評価点 x₁ 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、x₁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、一次元最 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、 u it_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 1	Etot -63.213925	de 591 0.00	1E 00E+00 0	resid .7951E-07	rsf 0.2444E-01	frgr -0.1955E+00	dx/at 0.0000E+00
の後、							
0	0.000000	-0.6321392	6E+02 -0	.195492178	E+00		
1	0.080000	-0.6322824	6E+02 -0	.16235414	E+00		
2	0.160000	-0.6323989	6E+02 -0	.128864728	E+00		
3	0.240000	-0.6324885	2E+02 -0	.949861278	2-01		
4	0.320000	-0.6325507	8E+02 -0	.60610679E	2-01		
5	0.400000	-0.6325853	0E+02 -0	.256181898	2-01		

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 Hellnamm-Feynman 力成分 f_s となっている。 $u - E_{tot}$ 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(max-IterOatom)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 (maxIterOcell)
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が 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。 ー回の計算で十分な格子緩和が行われなかった時は、Resume cell relaxation を1とセットして再計算する。このとき前の計算で達成された格子定数、原子位置の情報はpwm_*.varで格納され、そこから読み込まれるので、結晶データファイル*.primを書き換える必要はない。

6.2.2 結果の読み方

ここでは、nontrivial な例として三方晶の α ホウ素 (空間群 D_{3d}^5)を見てみる。初期の 格子定数は、

LATTICE PARAMETERS in a.u. 9.5563443 9.5563443 9.5563443 0.5290309 0.5290309 0.5290309

と実験値を与えている。

それに対するストレスは

total	stress (Ry,	/Bohr ³)		
	S(1,1)=	-3.88638E-04	S(2,3)=	2.30327E-10
	S(2,2) =	-3.88636E-04	S(3,1) =	-3.87469E-18
	S(3,3) =	-9.38125E-04	S(1,2) =	-3.37167E-18

という結果を与えている。

結果は今の場合、ファイルpwm_*.sumを見るのが良い。その中で

===== LATTICE MODIFICATION ================

の見出しを見つける。その後に格子定数最適化の条件が記述されている。

use Conjugate-Gradient Method convergence criterion for stress 2.000E-05 Max iteration 3 Symmetry constraints on the cell 3 0: triclinic, 1: monoclinic, 2: orthorombic, 3: tetragonal, 4: cubic under pressure = 0.00000 (GPa) = 0.000E+00 (Ry/Bohr^3)

その後に続くのが格子定数に関する共役勾配法の計算過程である。

IC	Etot0	Etot	del_E	epmin	res stress	its sea comp
0	-68.47157622	-68.47157622	0.000E+00	0.000E+00	0.109E-02	0.641E+00
1	-68.47157622	-68.47982412	-0.825E-02	0.248E-01	0.328E-04	0.132E-01
2	-68.47982412	-68.47983356	-0.944E-05	0.781E-03	0.244E-05	-0.643E-03
3	-68.47983356	-68.47983356	0.000E+00	0.781E-03	0.244E-05	0.138E-02
it	ceration stop					
beca	ause the change	in E 0.00E+00				
	became less th	an that allowed b	oy RSstress 2.0	00000E-05		
<<<<	<<<< final resu	lts <<<<<<<<		<<		
STEF	P Etot	delE	resid	rsf	rfm	
100	-68.47983356	-0.8715E-08	0.7725E-10	0.5726E-04	0.2039E-03	
Etot	c= −6.84798336E+	01 (Ry/cell)				
	-9.31717578E+	02 (eV/cell)				

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	
	9.38	3527485	9.38527485	9.38527485
	0.52	2019576	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=	0.0000000	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=	0.0080000	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うふうにエネルギーが変化する

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

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl} \tag{6.5}$$

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 $\Delta \sigma \approx 10^{-6} [\text{Ry/Bohr}^3]$ 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 $6.8 \times 10^{-3} [\text{Ry/Bohr}^3]$ 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変化す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したとき、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。

6.2.5 関係するオプション

- steepest_atom_relax ON
- steepest_cell_relax ON

原子位置にしろ、格子パラメータにしろその最適化では、デフォルトでは共役勾配法が適 用されるが、自由度の数が小さいときは必ずしもそれが一番有効的な方法というわけでは ない。ときとして最急勾配法がより良い場合もある。その時は上のように最急勾配法を選 ぶことが出来る。

• pressure_GPa=

1

格子パラメータの最適化の際、外部圧を加えたいとき用いる。

• trial_step0atps= 0.001

原子位置の最適化の一次元最小化において、放物線近似をするための試行変位の大きさ u_1 。ボーア単位。

• trial_step0cell=

0.001

格子パラメータの最適化の一次元最小化において、放物線近似をするための試行歪みの大きさ ε_1 。

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ブリルアン・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のブリルアン・ゾーンは図7.1のようになっていることを知っていな いといけない。それは文献[24]などを参照していただく。ブリルアン・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 のブリルアン・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。計算の後のデータを表示するためのプログ ラム ayband、pdosdr は柳瀬章の作成によるものを用いている(入力部だけ Osaka2k に 合わせている)。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 カファイル 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



図 7.2: pwbcd 計算プログラムの流れ

```
JobType
dos
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グ用であるので上の例のままで良い。

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、

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

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・ゾーンの外にはみ出たものは、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。

pwbcd 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ーオプションを付けるとき、 オプティマイズ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

の後には計算条件がリストされる。それに続いて見出し

の後には k 点サンプリングに関する情報があるで、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			с				р		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	Х	0	0	2/	2	2	2	0/	4	0.1949	3	0.37500	2
s	um ch	leck o	ver	wtk	=	1.	000	0000					
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 サプ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。

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 0 0 0/4 Conduction Bottom: 7.1785 at 3 Gap= 0.8889 from 0 0 0/4 to 2 2 0/4 final distribution of the valence bands IP2 - IP3 kn E2 E3 0cc2 - 0cc3-3 3 - 4 3.17028 -3.17028 2.00000 - 2.00000 1 2 2 3 - 4 4.91918 -4.91918 2.00000 -2,00000 6.28963 - 6.28963 2 - 4 2.00000 -2.00000 З 1 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本のうちいくつかが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 pwbcd の出力ファイル fort.2 を読み込んでこのことを実行する。² 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。³

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

¹この点に関して柳瀬先生から御教示いただいた。

²最新版は pdosdr である。これに関しては、スピンを入れた場合への拡張など記すべき事項が多くある。 それらの詳細は部分 DOS の描き方との関係で Technical Report No. 61 を参照されたい。

³この点での詳細は、Technical Report No. 83「Osaka2k における pdosdr の使い方 –*k* 点メッシュの切 リ方–」、さらに改良された Technical Report No. 114「DOS およびフェルミ面プロットにおける逆格子空 間メッシュ」を参照のこと。



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。

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
 不合理だがどうしょうもない。 4

使い方は、まず入力ファイル pdos.inp 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る。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ない。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

⁴この点は改善された。Technical Report No. 114「DOS およびフェルミ面プロットにおける逆格子空間 メッシュ」を参照のこと。

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

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 に持っていく。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りない。

7.1.4 フェルミ面

金属の場合は、pwbcd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、 pwbcd 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で、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 には、A1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。

図 7.4 は Mathematica で描かせたもので、上で述べたような不満があるが、そのよう な要求を満たすものとして、やはり柳瀬によるユーティリティTPERSP がある。これを Osaka2k でも利用できるようにした。詳しくは Techinical Report No. 104「フェルミ面の 描き方 TPERSP の使い方」、同じく No. 112「フェルミ面の実例、Pd」を参照のこと。



図 7.4: Alのフェルミ面。a) バンド2。(b) バンド3。

7.2 バンド構造

pwbcd 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 プロッティ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 などで読むことができる。

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) KB(3), ICB (in prim) 0 0 0 G 1 3 3 6 8 Κ 1 1 2 2 Х 2 W 1 3 4 1 1 1 2 L 0 0 0 G 1 0 2 1 Х 1 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)
```

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。

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

ここで一番重要な入力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

 $A \to B \to C \cdots$

となり、ノードの数が NKPTS で、以下ノードの $k \leq M \operatorname{KB}(3)/\operatorname{ICB}$ 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。⁵NDIV 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 りの電子数くらいで良いが、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。

 $^{{}^5}$ 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、 $\Gamma \to K \to X(1,1,0) \to W \to L \to \Gamma \to X(1,0,0)$ と取ることである。

手っ取り早くバンド図を見たければ、bnd_*.tblを二次元プロッティ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ティングは容易で 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。 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: Si 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」というものがあって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なら」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。次がスピンに関係した制御数(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 ayband.inp に書き込まれる。その中身は

-ORBIT						
				NLCOMP	NSPIN	IFILE
				JPR	JMARK	IPOINT(character)
2 100.0) 15	50.0)			
18	0	0	0	8 LD		
8 (8	0	0	8 DT		
8 (8	4	0	8 W		
8 (6	6	0	8 K		
8 (0	0	0	8 SM		
				Fermi	level	
Potent	tial	L				
	-ORBIT 2 100.(1 8) 8) 8) 8) 8) 8) 8	-ORBIT 2 100.0 19 4 8 0) 8 8) 8 8) 8 6) 8 0 DPotential	-ORBIT 2 100.0 150.0 4 8 0 0) 8 8 0) 8 8 4) 8 6 6) 8 0 0 DPotential	-ORBIT 2 100.0 150.0 4 8 0 0 0 0 8 8 0 0 0 8 8 4 0 0 8 6 6 0 0 8 0 0 0	-ORBIT NLCOMP JPR 2 100.0 150.0 4 8 0 0 0 8 LD 0 8 8 0 0 8 DT 0 8 8 4 0 8 W 0 8 6 6 0 8 K 0 8 0 0 0 8 SM Fermi	-ORBIT NLCOMP NSPIN JPR JMARK 2 100.0 150.0 4 8 0 0 0 8 LD 0 8 8 0 0 8 DT 0 8 8 4 0 8 W 0 8 6 6 0 8 K 0 8 0 0 0 8 SM Fermi level DPotential

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 が機械的にかっ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、閃亜鉱型の 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っつくが、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と、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⁶。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

$$D(\alpha) = \exp(i\mathbf{k} \cdot \tau_{\alpha})\Gamma(\alpha) \tag{7.1}$$

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、∆ 軸上では、サブルーチン TSIRDS は

TMB_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 22+-JI0000

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。

⁶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 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 の既約表現委号を示す

I DI A D D D M M M A D C M y 。							_								
	D	E	C_2	$2C_4$	$2\sigma_{\rm v}$	$2\sigma_{\rm d}$		Γ	E	C_2	$2C_4$	$2\sigma_{\rm v}$	$2\sigma_{\rm d}$		
				au	au						au	au			
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		
							-		$\eta = \exp[-i\mathbf{k}\cdot\tau]$						

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。 $k \to 0$ 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。表現が一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

IMR NO 1 DIMENSION= 2 1 2 3 4161819222526272840424346 1 1 + + + + 0 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 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。 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。

40 0 0/ 40 X 2 11 1 th point = 1,0000 0,0000 0,0000 Characters #IR= 1 ND = 2MG= 16 3 18 JG= 1 2 4 16 19 22 25 26 27 28 $Im= 0.00 \ 0.0$ JG= 40 42 43 46

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}	$2C'_2$	$2\sigma_{\rm d}$	Γ	E	C_{2x}	$2C'_2$	$2\sigma_{\rm d}$
			au					au		
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

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 波動関数のプロット

実空間で波動関数をプロットさせたいときがある。Osaka2k では2種類の方法を提供する。それらは[42]を参照。

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。どちらにしろこれらの 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 (Å 単位で amax) に取られる。

- 基本単位格子は三つの基本並進ベクトル {t₁, t₂, t₃} で表される(ノートブック中で { t1, t2, t3})
- ・ 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 {*gp*1, *gp*2, *gp*3} 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 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

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。

lattice[0]=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 まうと、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ーを 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

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の点に取りたければ、

$$\overrightarrow{\text{pl0}} = \frac{3\overrightarrow{\text{gp0}} + \overrightarrow{\text{gp3}}}{4} \tag{A.1}$$

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。 それの例が図 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 はデーター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

$$N_{el} = \sum_{i=1}^{NA3} \rho_i \times \frac{1}{NA3} \tag{A.2}$$

と取ってある。Nel 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ない。



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{1,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、 藤田洋士、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 Stete 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 Stete 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. Aschcroft 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, Camridge, 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. Jepson and O. K. Anderson, Solid State Commun. 9, 1763 (1971)
- [37] C. Herring, Phys. Rev. **52**, 365 (1937)
- [38] G. F. Koster, Solid Stete 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 adn 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] Technical Report No. 21 「波動関数のプロット」
- [43] D. Frenkel and B. Smit, "Understanding Molecular Simulation", (Academic, San Diego, 1996)
- [44] S. E. Koonin and D. C. Meredith, "Computational Physics", (Addison Wesley, Massachusets, 1990)