

文部科学省次世代 I T 基盤構築のための研究開発  
「イノベーション基盤シミュレーションソフトウェアの研究開発」

CISS フリーソフトウェア

ナノ・物質・材料・マルチスケール機能シミュレーション

PHASE ver.10.01

## ユーザマニュアル

本ソフトウェアは文部科学省次世代 I T 基盤構築のための研究開発「イノベーション基盤シミュレーションソフトウェアの研究開発」プロジェクトによる成果物です。本ソフトウェアを無償でご使用になる場合「CISS フリーソフトウェア使用許諾条件」をご了承頂くことが前提となります。営利目的の場合には別途契約の締結が必要です。これらの契約で明示されていない事項に関して、或いは、これらの契約が存在しない状況においては、本ソフトウェアは著作権法など、関係法令により、保護されています。

お問い合わせ先

(契約窓口)

(財) 生産技術研究奨励会

〒 153-8505 東京都目黒区駒場 4-6-1

(ソフトウェア管理元) 東京大学生産技術研究所 革新的シミュレーション研究センター

〒 153-8505 東京都目黒区駒場 4-6-1

FAX: 03-5452-6662

E-mail: [software@ciss.iis.u-tokyo.ac.jp](mailto:software@ciss.iis.u-tokyo.ac.jp)

---

## **COPYRIGHT of the program codes**

Copyright (C) 1993-2006 Hideki Katagiri, Koichi Kato, Tsuyoshi Miyazaki, Yoshitada Morikawa, Hideaki Sawada, Toshihiro Uchiyama, Tsuyoshi Uda, Takahiro Yamasaki, Noriaki Hamada, Akira Yanase, Takenori Yamamoto, Hideaki Tsukioka, Masakuni Okamoto, Hideo Mizouchi, Kiyoshi Betsuyaku and Kazuki Mae.

It is understood by the authors that the Institute of Industrial Science (IIS), the University of Tokyo, distributes this program as "CISS Free Software" with users' agreement with the terms and conditions written in the file, LICENSE.pdf or LICENSE\_J.pdf (in Japanese).

## **HISTORY**

The original version of this set of the computer programs "PHASE" was developed by the members of the Theory Group of Joint Research Center for Atom Technology (JRCAT), based in Tsukuba, in the period 1993-2001. The names of the contributors to the original version are Hideki Katagiri, K. Kato, T. Miyazaki, Y. Morikawa, H. Sawada, T. Uchiyama, T. Uda and T. Yamasaki. Since 2002, this set has been tuned and new functions have been added to it as a part of the national project "Frontier Simulation Software for Industrial Science (FSIS)", which is supported by the IT program of the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan. The program was developed further mainly by T. Yamasaki. T. Uda, T. Yamamoto, H. Tsukioka, M. Okamoto, H. Mizouchi, K. Betsuyaku and K. Mae contributed to the improvement of the code. The tetrahedron interpolation codes developed by N. Hamada, A. Yanase and Kiyoyuki Terakura was included. The symmetrization code developed by A. Yanase and N. Hamada was also included. The manual and tutorial were written by Makoto Itoh with the cooperation by Mineo Saito, H. Tsukioka, T. Yamamoto and T. Yamasaki. The sample calculations were prepared by T. Yamamoto, H. Tsukioka and Hiroyoshi Momida. Since 2006, this program set has been developed as a part of the national project "Revolutionary Simulation Software (RSS21)", which is supported by the next-generation IT program of MEXT of Japan. Since 2008, this program set has been developed as a part of the national project "Research and Development of Innovative Simulation Software", which is supported by the next-generation IT program of MEXT of Japan. The activity of "Multiscale Simulation System for Function Analysis of Nanomaterials", CISS, is supervised by Takahisa Ohno.

## **CONTACT ADDRESS**

Center for Research on Innovative Simulation Software The Institute of Industrial Science (IIS), The University of Tokyo

4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan

FAX +81-(0)3-5452-6662

E-mail: [software@ciss.iis.u-tokyo.ac.jp](mailto:software@ciss.iis.u-tokyo.ac.jp) URL <http://www.ciss.iis.u-tokyo.ac.jp>

\* When distributing CISS Software duplications, the user must attach the full text in this file.

---

License to Use CISS Free Software for noncommercial purposes  
Terms and Conditions of the CISS Free Software License

The Center for Research on Innovative Simulation Software (CISS) at the Institute of Industrial Science, the University of Tokyo gives explicit permission for anyone to use any or all of the free software that is maintained and made publicly available at the CISS site free of charge, subject to the terms and conditions detailed below.

1. Definition of CISS Free Software

CISS Free Software is any software explicitly marked “ CISS Free Software ” in CISS project source programs, object programs, specifications, design specifications, data, implementation results, and instruction manuals.

2. Extent of Free Use

Users may use CISS Free Software free of charge to run their own data, and use any results obtained for their own personal use. Users also have the rights to copy, to modify, and to redistribute the CISS Free Software.

3. Rules for Modification and Distribution

If the user creates a modified version of CISS Free Software by modifying the software itself, by incorporating it into other software, or any other means; then copies and/or distributes the software, the user must retain the words “ CISS free software ” in the name of the modified version (e.g., if the CISS free software is named ProteinDF, the new software is named \_\_\_\_\_/ProteinDF.); however, this shall not apply if the user concludes separately a contract for the purpose of profit-making business. And also the user displays a copyright notice in the modified version.

The “ copyright notice ” in the internal code of the CISS Free Software may not be altered for any reason, except to update or add to modification records such as altering the name of the modifier or the date of modification.

4. Copyright Notice

Users must prominently and conspicuously display the copyright notice in every CISS Free Software copy at or near the beginning of the credits along with the name of the software, the version, and the copyright holder. When distributing copies of CISS Free Software, the user must attach the full text of these Terms and Conditions without any changes.

5. User Obligations

To publicly acknowledge that results have been achieved using CISS Free Software, users are obligated to clearly display the name, version, and copyright holder, and acknowledge that “these results were achieved by using Innovative Simulation Software for an Industrial Science Project. ”

If the user modifies the CISS Software and acknowledges that results were achieved using the software, the user must attach an explanation detailing how the software was modified.

We request that users report any bugs or problems they discover in using the CISS Software to the Center for Research on Innovative Simulation Software at the Institute of Industrial Science, the University of Tokyo. Users may not publicly announce or disclose bugs or problems they discover in CISS software without permission.

6. Commercial Use

If a user intends to use CISS Free Software for a commercial purpose such as described in examples (1)-(3) below, the user must enter into a separate commercial license agreement before using the CISS software.

(1) A user copies and distributes CISS Free Software, then demands compensation from the recipient for the software itself as a copyrighted product or for copying and distributing the software.

(2) A user (corporate or individual) uses CISS Free Software not for personal use but to provide services to other parties, regardless of whether the services are offered gratis or for a fee.

(3) A user seeks to assume a right of pledge, a security interest, or some other form of commercial interest in CISS Free Software, including portions of the software that were modified by the user.

However, if a public entity seeks to provide services using CISS software for the purpose disseminating the software, we require an exchange of memorandums between the CISS and the entity (in lieu a conventional for-profit license agreement) detailing the nature of the service, regardless of whether the proposed service is offered gratis or for a fee. The user acknowledges in advance that if he or she violates any of the provisions of this agreement, the copyright holder of any software shall prohibit the user from using the software. The user also acknowledges in advance that the copyright holder is entitled to be compensated by an amount equivalent to any profit gained by the user through the violation of the terms of this agreement.

7. No Warranty

The Institute of Industrial Science (IIS), the University of Tokyo, the Foundation for the Promotion of Industrial Science, and other concerned parties disclaim all warranties with respect to the quality, the performance,

or the results of CISS Free Software, either express or implied. The user assumes sole responsibility for the use of CISS software including any damages or losses arising out of the use of the CISS software.

#### 8. Violations of Terms and Conditions

If a user is found to be in violation of these Terms and Conditions, he or she agrees to immediately pursue any and all steps required by the Institute of Industrial Science, the University of Tokyo to get back into compliance.

## CISS フリーソフトウェア使用許諾条件

東京大学生産技術研究所 革新的シミュレーション研究センター（以下 革新センター）は、次の条件や制限のもとで、革新センターで管理・公開するプロジェクト等による成果物の全てまたは一部を無償で使用することを許諾します。

1. CISS フリーソフトウェアの定義革新センター（CISS）で管理しているソースプログラム、オブジェクトプログラム、仕様書、設計書、データ、実行結果 および マニュアルなどの内、インターネット上に公開しているソフトウェアを CISS フリーソフトウェアと呼びます。

2. 無償使用の範囲利用者が CISS フリーソフトウェアを無償で利用できる行為には、自己のために CISS フリーソフトウェアを任意のデータを用いて実行する行為、その結果を利用者の自己のために使用する行為、CISS フリーソフトウェアを複製し頒布する行為、および、CISS フリーソフトウェアを改変しそれを実行する行為等を含みます。

3. 改変・頒布での遵守事項 CISS フリーソフトウェアを変更したり、他のソフトに組み込む等の行為により、改変した CISS フリーソフトウェアを複製・頒布する場合は、そのソフトウェア名には CISS フリーソフトウェアの名称を残して（例えば、CISS フリーソフトウェアの名称を ProteinDF とした場合、ProteinDF のようにネーミング）下さい。ただし、別途営利目的の場合における実施許諾契約を締結している場合はこの限りではありません。また、著作権表示を行うことを義務づけます。目的の如何を問わず、CISS フリーソフトウェア内部コードの『著作権表示』記載部分を修正する行為は、改変者氏名や改変日時などの改変記録を追加する場合を除き、禁止されています。

4. 著作権の表示利用者は、各々の CISS フリーソフトウェアの複製物に、ソフトウェア名・バージョン・著作者氏名などの著作権表示を表示の先頭部等の箇所に適切かつ目立つように掲載するとともに、頒布する場合は、複製物に本許諾条件の全文をそのまま添付しなければなりません。

5. 利用者義務 CISS フリーソフトウェアを利用した結果を公表する場合には、関連プロジェクト等の成果を利用した（例：『革新的シミュレーションソフトウェアの研究開発プロジェクトの成果を利用した』）旨を、使用した CISS ソフトウェアの名前、バージョン、著作者氏名などの記載とともに、明示して下さい。利用者が CISS ソフトウェアを改変し、その実行結果を公表する場合は、改変内容や改変履歴が特定できる説明を添付して公表しなければなりません。利用者が CISS ソフトウェアのバグや不具合を発見した場合、革新センターに報告して下さい。発見したバグや不具合を許可なく公表したり、第三者に知らせることを禁止します。

6. 営利目的に使用する場合利用者は、CISS フリーソフトウェアを下記 (1)～(3) に例示するような営利目的に使用する場合には、事前に別途営利目的の場合における実施許諾契約を締結する必要があります。

(1) 利用者が CISS フリーソフトウェアを複製・頒布する場合、著作物としての対価のみならず、複製ないし頒布に必要な経費など経済的価値を、頒布を受ける者に対して提示ないし要求すること。(2) 法人を含み利用者は、自己の目的に限り CISS フリーソフトウェア実行が許諾されているものであり、有償無償を問わず第三者へのサービスのために CISS ソフトウェアを実行する行為をすること。(3) 利用者は、自己が改変した部分も含み、CISS フリーソフトウェアを質権や担保など、いかなる商取引の対象に加えること。

ただし、公的機関が当該ソフトウェアの普及促進を目的としてそれを利用したサービスを提供する場合は、そのサービスの有償無償を問わず、別途その内容に関して革新センターとの間で覚書等を交わすことをもって営利目的用実施許諾契約締結の代用とすることができるものとします。利用者が本項に反する行為を行った場合には、各ソフトウェア等の著作権者によりその利用を差し止められることを利用者は予め了解します。かつ、利用者は、それにより得た利益相当額の賠償をもとめられることも予め了解します。

7. 無保証 CISS フリーソフトウェアは、その品質や性能あるいは実行結果について、利用者に対してはいかなる保証もされていません。利用者は自己の責任において使用することに同意することとし、もし使用することにより損害が生じた場合には、第三者への損害や被害の修復も含み、その結果責任は全て利用者に帰することとします。

8. 利用者が本使用許諾条件に違反した場合利用者が本使用許諾条件に違反した場合には、利用者は、革新センターがその状態を是正するために必要と認めて行う措置に無条件に従うものとします。

- 以上 -

## 目次

<b>1</b>	<b>はじめに</b>	<b>1</b>
1.1	PHASE の特徴	1
1.2	本マニュアルのキーワードリスト	2
1.3	更新履歴	2
<b>2</b>	<b>PHASE における作業の流れ</b>	<b>4</b>
2.1	PHASE のフローチャート	4
2.2	PHASE 関連プログラム	4
2.3	入出力ファイル	4
<b>3</b>	<b>入力ファイル</b>	<b>7</b>
3.1	FINP	7
3.2	識別子の説明	11
3.3	FPOT(n)	18
<b>4</b>	<b>計算例 1 : Si 結晶の全エネルギー計算 (Si<sub>8</sub>)</b>	<b>20</b>
4.1	入力データ	20
4.2	計算の実行と途中経過	23
4.3	出力データ	23
4.4	強制終了の方法	25
4.5	再計算の方法	25
4.6	交換相関エネルギーの型の指定方法	26
4.7	$k$ 点の指定方法	26
4.8	Brillouin 領域の積分方法	26
4.9	2 種類以上の原子を扱う方法	26
4.10	原子座標を原子単位系で入力する方法	27
4.11	反転中心を考慮する方法	27
4.12	分子動力学で指定原子を固定する方法	28
<b>5</b>	<b>計算例 2 : 対称性を考慮した計算 (Si<sub>2</sub>)</b>	<b>29</b>
5.1	単位胞の指定 : Si <sub>2</sub> の場合	29
5.1.1	単位胞を基本格子で指定	29
5.1.2	単位胞を格子定数で指定	29
5.2	対称性の指定	30
5.2.1	結晶構造を入力する方法	30
5.2.2	生成元を入力する方法	32
5.2.3	対称操作を自動的に決定する方法	32
5.3	反転対称性がある場合	33
5.3.1	指定しないとき	33
5.3.2	指定するとき	33
<b>6</b>	<b>計算例 3 : スピン分極がある場合 (Fe)</b>	<b>34</b>
6.1	スピン自由度の指定方法	36
6.2	スピン分極を固定する方法	36
6.3	スピンを考慮している場合の SCF 計算の収束性	36
6.4	スピン分極の変化	37
6.5	cube ファイル形式による電荷密度分布の出力	38
6.6	状態密度 と バンド構造	38
<b>7</b>	<b>計算例 4 : 水素終端 Si(001) 表面</b>	<b>40</b>

<b>8</b>	<b>PHASE における最適化計算</b>	<b>43</b>
8.1	作業フロー	43
8.2	電子状態最適化ループ	43
8.3	波動関数 solver と電荷密度混合のオプション選択の仕方	44
8.3.1	波動関数 solver	44
8.3.2	電荷密度混合	45
8.3.3	収束を加速させるテクニック	45
8.4	SCF 計算ベンチマークテスト	48
8.4.1	FCC-Cu	48
8.4.2	Fe(100) 表面	52
8.5	構造最適化ループ	54
8.5.1	quenched MD 法	55
8.5.2	cg 法	55
8.5.3	gdiis 法	55
8.5.4	bfgs 法	55
<b>9</b>	<b>分子動力学シミュレーション</b>	<b>57</b>
9.1	はじめに	57
9.2	小正準集合	57
9.2.1	入力の説明	57
9.2.2	出力結果	59
9.3	正準集合	60
9.3.1	入力の説明	60
9.3.2	熱浴の質量の設定方法	61
9.3.3	出力の解析	61
<b>10</b>	<b>振動解析</b>	<b>63</b>
10.1	はじめに	63
10.2	振動解析の理論	63
10.3	入力の説明	64
10.3.1	原子座標と対称性の入力	64
10.3.2	元素の質量の指定	64
10.3.3	原子変位の選択	64
10.4	実行の方法	64
10.5	振動解析結果の出力ファイル mode.data	64
10.6	力データファイル force.data の入出力	65
10.7	例：水分子の振動解析	65
10.7.1	構造最適化	65
10.7.2	振動解析	67
<b>11</b>	<b>STM 像のシミュレーションプログラム</b>	<b>70</b>
11.1	計算手法	70
11.2	プログラムについて	74
11.2.1	ソースプログラム構造	74
11.2.2	入出力ファイル	75
11.3	計算の実際	76
11.3.1	入力ファイル F_INP の作成について	76
11.3.2	実行方法	77
11.3.3	計算例	78
<b>12</b>	<b>電子状態解析：局所状態密度と部分電荷密度</b>	<b>79</b>
12.1	はじめに	79
12.2	原子分割局所状態密度	80
12.3	層分割局所状態密度	80
12.4	部分電荷密度	82

<b>13 DFT+U 法</b>	<b>85</b>
13.1 はじめに	85
13.2 理論	85
13.3 使い方	85
<b>14 最大局在ワニエ関数</b>	<b>89</b>
14.1 はじめに	89
14.2 計算方法	89
14.3 計算例:Si の最大局在ワニエ関数	89
<b>15 陽電子寿命計算</b>	<b>91</b>
15.1 はじめに	91
15.2 計算手法	91
15.3 具体的なやり方: Si 結晶の計算例	92
15.3.1 input file の解説	92
15.3.2 output file の解説	93
15.3.3 cube file の解説	93
15.4 信頼性のある計算を行うために	93
15.4.1 擬ポテンシャルの選択	94
15.4.2 カットオフエネルギーの選択	94
15.4.3 陽電子波動関数の収束の確認	94
<b>16 NEB 法</b>	<b>95</b>
16.1 はじめに	95
16.2 理論	95
16.2.1 作用力	95
16.2.2 ばね定数の計算方法 [4]	95
16.3 使い方	96
16.3.1 実行方法	96
16.3.2 入力データ例	96
16.3.3 NEB 関連ファイル	98
16.3.4 入力データ (NEB 関連のキーワード)	98
16.4 例題	102
16.4.1 入力ファイルの説明	102
16.4.2 計算結果	104
<b>17 拘束条件付きダイナミクスと Blue Moon 法</b>	<b>106</b>
17.1 はじめに	106
17.2 拘束条件付きダイナミクスの基本的な使い方	106
17.2.1 入力ファイルの記述方法	106
17.2.2 計算の実行方法	108
17.2.3 出力ファイル	109
17.3 複数の反応座標を逐次変化させる方法	109
17.4 ファイルから反応座標の変化の仕方を指定する方法	110
17.5 Blue Moon 法によって自由エネルギーを計算する方法	111
17.6 例題	113
<b>18 Meta-dynamics 法</b>	<b>115</b>
18.1 はじめに	115
18.2 理論	115
18.2.1 概要	115
18.2.2 運動方程式のもととなるハミルトニアン	116
18.3 利用方法	116
18.3.1 入力ファイルの記述	116
18.3.2 計算の実行方法	120
18.3.3 出力ファイル	120



18.4 例題	122
18.4.1 計算方法	122
18.4.2 計算結果	124
<b>19 非局所相関項</b>	<b>127</b>
19.1 はじめに	127
19.2 非局所相関項の理論 (ポスト計算用)	127
19.2.1 理論概要	127
19.2.2 入出力の説明	129
19.2.3 積層グラフィットの全エネルギー計算	131
19.3 非局所相関項を含めた電子状態計算 (セルフ・コンシステント計算用)	131
19.3.1 理論概要	131
19.3.2 入出力の説明	132
<b>20 ハイブリッド汎関数法</b>	<b>133</b>
20.1 概説	133
20.2 使用方法	133
20.3 計算例	134
20.3.1 水素分子	134
20.3.2 Si 結晶	134
<b>21 PAW 法</b>	<b>136</b>
21.1 利用方法	136
21.2 計算例	136
<b>22 Q and A</b>	<b>138</b>
<b>23 終りに</b>	<b>139</b>

## 表 目 次

1	入出力ファイル一覧	6
2	入力ファイル F_INP に使用される識別子とその説明	11
3	ブラベー格子と晶系	30
4	ブラベー格子の基本並進ベクトル	31
5	波動関数ソルバーによる計算時間の比較	51
6	Fe (100) 表面の収束に至るまでの SCF 計算回数と到達した全エネルギー	54
7	構造緩和法の違いによる収束にいたる回数の比較	56
8	振動解析に関係する変数の説明	69
9	占有行列の添え字と軌道の性格の対応表	87
10	NEB で利用できるファイルポインター	99
11	NEB に関連する入力データ	99
12	収束判定条件設定のパラメーター	100
13	PAW 法および US 法によってもとめた格子定数と体積弾性率	137

## 図 目 次

1	処理過程のフローチャート	5
2	Si 結晶の構造図	20
3	Si <sub>2</sub> の原子構造図	29
4	六方晶系と菱面体晶系の関係	31
5	Fe の電荷密度分布図	34
6	Fe の状態密度図	38
7	Fe のバンド構造	39
8	水素終端した Si(001)- <i>p</i> (2 × 1) 表面の構造図	40
9	最適化計算のフローチャート	43
10	波動関数ソルバーによる収束の速さの比較。	51
11	内部エネルギー，運動エネルギー，全エネルギーと時間の関係。	60
12	瞬間的な温度と時間の関係。	62
13	水分子の構造	68
14	水分子の振動モードの振動数	68
15	水分子の振動モードの固有ベクトル	69
16	Si (2 × 1) の (100) 面の電子状態密度。	78
17	BaO/Si(001) 界面構造の原子分割の局所状態密度。	81
18	BaO/Si(001) 界面構造の層分割局所状態密度。	82
19	BaO/Si(001) 界面構造の部分電荷密度。	84
20	ハバード補正による軌道エネルギーの変化	85
21	SrTiO <sub>3</sub> の状態密度	88
22	(a) Si 結晶と (b) GaAs 結晶の最大局在ワニ関数	90
23	Si 結晶中の価電子分布 (a)，陽電子分布 (b)，ならびに，電子・陽電子対分布 (c)。	93
24	本例題の始状態。	102
25	本例題の終状態。	102
26	NEB 力の履歴。	104
27	最終的に得られる反応経路と各レプリカのエネルギーの関係。	104
28	遷移状態における原子配置。	105
29	H <sub>2</sub> O <sub>2</sub> および H <sub>2</sub> S <sub>2</sub> 分子の分子構造。	113
30	H <sub>2</sub> O <sub>2</sub> および H <sub>2</sub> S <sub>2</sub> 分子の 2 面角とエネルギーの関係。	114
31	Meta-dynamics シミュレーションの模式図。	115
32	C <sub>4</sub> H <sub>6</sub> 分子の分子構造	122
33	C <sub>4</sub> H <sub>6</sub> 分子の分子構造	123
34	本計算例によって得られる C <sub>4</sub> H <sub>6</sub> 分子の自由エネルギー表面。	124
35	2 面角とバイアスポテンシャル更新回数の関係	125
36	炭素原子間距離とバイアスポテンシャル更新回数の関係	125

37	Meta dynamics シミュレーションによって得られた分子構造のスナップショット . (a) : バイアスポテンシャル 2 回更新 (b) バイアスポテンシャル 690 回更新 (c) バイアスポテンシャル 1,500 回更新 (d) バイアスポテンシャル 18,070 回更新 . . . . .	126
38	本ルーチン (vdW) の計算の流れ . . . . .	129
39	積層グラファイトの全エネルギーの層間距離依存性 . GGA (赤線) と GGA + vdW (緑線) による比較 . . . . .	132
40	PBE 汎関数法, HSE06 汎関数法, PBE0 汎関数法, HF 法による水素分子の HOMO 準位。 . . . .	134
41	HSE06 汎関数法による Si 結晶のバンド構造。 . . . .	135
42	クロムの <i>EV</i> 曲線。赤線が PAW の結果, 緑線が US の結果。各手法で得られた最も低いエネルギーをエネルギーの原点としている。 . . . .	137

1 はじめに

本マニュアルでは、PHASE の詳細な使用法についてご説明いたします。本マニュアルを読む前に別冊のチュートリアルをお読みいただき、簡単な例題を体験されていることを前提として説明します。

1.1 PHASE の特徴

チュートリアルの冒頭にも書いてありますが、PHASE で可能な計算や PHASE の特徴を、最初に簡単にまとめておきます。

PHASE で遂行可能な計算
全エネルギー
電荷密度の分布
電子の状態密度 (DOS)
バンド構造
安定な原子構造
振動解析
STM 像シミュレーション

PHASE の特徴
Local Density Approximation (LDA) や Generalized Gradient Approximation (GGA) を使った、密度汎関数法に基づいた電子状態の第一原理計算プログラム。この方法を使った計算結果がさまざまな学術雑誌に掲載されている、非常に信頼性の高い計算法。
イオンコアの影響を擬ポテンシャルによって取り込み、価電子の波動関数を平面波により展開。Troullier–Martins 型のソフト擬ポテンシャルと、Vanderbilt 型のウルトラソフト擬ポテンシャルが使用可能。これらのソフト擬ポテンシャルを用いることで、計算に必要な平面波の数を減らすことができ、大規模計算が可能
波動関数の最適化には、残差最小化 (RMM) 法、Steepest Descent (SD) 法、Modified SD 法、1 次元探索法などが選択可能。電荷密度の混合には、単純混合法、Broyden 法などが使用可能
幾何学的構造の最適化法としては、Quenched MD 法、GDIIS 法、CG 法が使用可能
入力ファイルをタグ形式にして、初めて使用するユーザーでも、入力パラメータの物理的意味が理解し易いように構成
バンド構造、状態密度、電荷密度分布などの計算結果を表示するプログラムが付属 (使用可能な描画用ソフトウェアには、「革新的シミュレーションソフトウェアの研究開発」プロジェクトで開発された Biostation viewer 等があります)
MPI を 利用した並列計算が可能

## 1.2 本マニュアルのキーワードリスト

次に、本マニュアルで使用する略称のキーワード・リストを、一覧表にまとめておきます。

キーワード	説明
<b>PHASE</b>	本プログラムの本体. SCF 計算等に使用
<b>ekcal</b>	状態密度やバンドの計算に使用
<b>CIAO</b>	擬ポテンシャル生成用ソフトウェア. Code for Investigating Atomic Orbitals の略
<b>SCF</b>	self-consistent field method, 自己無撞着場の方法
<b>DOS</b>	density of states, 状態密度
<b>LDA</b>	Local Density Approximation, 局所密度近似
<b>GGA</b>	Generalized Gradient Approximation
<b>GGAPBE</b>	J. P. Perdew, K. Burke, and M. Ernzerhof, Physical Review Letters <b>77</b> , 3865 (1996) 参照
<b>GGAPW91</b>	Perdew-Wang 91, in J. P. Perdew, <i>Electronic Structure of Solids, '91</i> , edited by P. Ziesche and H. Eschrig (Akademic Verlag, Berlin, 1991), p.11, 参照
<b>LDAPW91</b>	John P. Perdew and Yue Wang, Physical Review B <b>45</b> , 13244 (1992) 参照
<b>Pulay</b>	P. Csaszar and P. Pulay, <i>Geometry Optimization by Direct Inversion in the Iterative Subspace</i> , Journal of Molecular Structure <b>114</b> , 31-34 (1984) 参照
<b>TM</b>	N. Troullier and J. L. Martins, Solid State Commun. <b>74</b> , 613(1990) 参照
<b>Ultrasoft</b>	D. Vanderbilt, <i>Soft Self-Consistent Pseudopotentials in a Generalized Eigenvalue Formalism</i> , Phys. Rev. B <b>41</b> , 7892-7895 (1990); K. Laasonen, R. Car, C. Lee, D. Vanderbilt, <i>Implementation of Ultrasoft Pseudopotentials in Ab Initio Molecular-Dynamics</i> , Phys. Rev. B <b>43</b> , 6796-6799 (1991) 参照
<b>GDIIS</b>	<i>Geometry optimization by Direct Inversion in the Iterative Subspace</i> , J. Mol. Structure, <b>114</b> : (MAR), 31-34 (1984), G. Fogarasi, X Zhou, P.W. Taylor and P. Pulay, Journal of the American Chemical Society <b>114</b> , 8191 (1992) 参照
<b>Quenched MD</b>	quench は急冷を意味する. MD は molecular dynamics (分子動力学) の略
<b>SD</b>	steepest descent, 最急降下法
<b>MSD</b>	modified steepest descent, 改良型最急降下法
<b>lm</b>	line minimization, 一次元探索
<b>RMM</b>	residual minimization method, 残差最小化法
<b>Biostation viewer</b>	「革新的シミュレーションソフトウェアの研究開発」プロジェクトで開発された描画用ソフト
<b>MPI</b>	Message Passing Interface 複数の演算装置を使って並列計算を行うための計算機プログラム

## 1.3 更新履歴

2009 年 6 月以降の更新履歴を以下に記述します。

- 2009/06 バージョン 8.00 公開
  - － 拘束条件付きの動力学を追跡する機能が追加されました。
  - － DFT+U 法による構造最適化/分子動力学シミュレーションに対応しました。
- 2010/03 バージョン 8.01 公開
  - － BLAS を利用した高速化に対応しました。
- 2010/06 バージョン 9.00 公開
  - － キャッシュチューニング、BLAS 化をさらに進め、高速化を行いました。
  - － ファンデルワールス相互作用を考慮することができるようになりました。

- 自由エネルギー計算が行えるようになりました。
- DFT+U 法を利用して、バンド構造の計算ができるようになりました。
- ハイブリッド汎関数が利用できるようになりました。
- 2011/06 バージョン 10.00 公開
  - 電子状態計算の収束性が向上しました。
  - PAW 法に対応しました。
  - メタダイナミクスに対応しました。
  - van der Waals 密度汎関数のセルフコンシステントな実装を行いました。
  - 構造最適化に BFGS 法が利用できるようになりました。
  - PHASE TOOLS に新たなスクリプトを加えました。
  - 擬ポテンシャルの読み込みに関する不具合を修正しました。  
この修正によって、従来のバージョンと全エネルギーの絶対値は一致しなくなります。
- 2011/08 バージョン 10.01 公開
  - スピンを考慮している系の収束性が向上しました。
  - GGA に関する不具合を修正しました。  
この修正によって、従来のバージョンと全エネルギーの絶対値は一致しなくなります。

## 2 PHASE における作業の流れ

### 2.1 PHASE のフローチャート

PHASE に関連する、入出力ファイルと各種プログラムの関係を、図 1 にまとめました。図中で、青色と赤色はそれぞれ、実行プログラムと入出力ファイルを指しています。また、実行プログラムに向かう矢印は入力を、逆は出力を意味しています。PHASE を使ってエネルギーを収束させた後、電荷密度の出力ファイルである F\_CHGT などを入力として、ekcal で 状態密度図やバンド構造を計算することができます。それらの出力ファイルである F\_DOS や F\_ENERG を基に、状態密度図やバンド構造図を描くことも可能です。さらに、cube ファイル形式で出力された電荷密度分布のファイル F\_CHR から「革新的シミュレーションソフトウェアの研究開発」プロジェクトで開発された描画用ソフトである Biostation viewer を使って電荷密度を 3 次元的に表示することもできます。

### 2.2 PHASE 関連プログラム

図 1 に示したように、PHASE の作業に関連するプログラムには、CIAO, Biostation Viewer, dos.pl, band.pl, それに band\_kpoint.pl があります。

- (1) CIAO : PHASE の入力に使う擬ポテンシャルを作成します。CIAO を使って擬ポテンシャルが作成されています。これらは、戦略 Pjweb にある周期表で各原子をクリックする事により、ダウンロードが可能です。
- (2) Biostation Viewer : PHASE で計算した cube ファイル形式の電荷密度分布を、このソフトで可視化できます。
- (3) dos.pl と band.pl : PHASE 計算結果の解析用に作成された Perl スクリプトです。前者は状態密度図の、後者はバンド構造図の、Postscript 形式によるファイルを生成します。
- (4) band\_kpoint.pl : バンド計算を行うための入力データ F\_KPOINT を作成します。
  - (1) は「戦略的基盤ソフトウェアの開発」プロジェクトのホームページ (<http://www.fsis.iis.u-tokyo.ac.jp/result/software/>) より別途ダウンロードして下さい。
  - (2) は「革新的シミュレーションソフトウェアの研究開発」プロジェクトのホームページ (<http://www.rss21.iis.u-tokyo.ac.jp/result/download/>) からダウンロードが可能です。
  - (3) と (4) は共に、phase パッケージ中の tools に同梱されています。

### 2.3 入出力ファイル

PHASE を

```
% mpirun ../bin/phase
```

の様に走らせると、作業ディレクトリー内に置かれている file\_names.data が最初に読み込まれます。このファイルは入出力ファイルを管理しています。例えば、samples というディレクトリーの下に Si8 にある file\_names.data の冒頭には、次のような記述があります。

```
&fnames
F_INP      = './input_scf_Si8.data'
F_POT(1)   = '../pp/Si_ldapw91_nc_01.pp'
F_DYNM     = './nfdynm.data'
F_CHGT     = './nfchgt.data'
F_ENF      = './nfefn.data'
F_ZAJ      = './zaj.data'
&end
```

この例から分かるように、file\_names.data には、PHASE の実行に必要なファイル名が割り当てられています。入出力ファイル file\_names.data を使ってユーザーが指定可能なファイルと、それらのデフォルト値は、表 1 のように構成されていますが、もし扱うファイル名を変更したければ、file\_names.data を編集して、ファイル名を適宜変更して下さい。

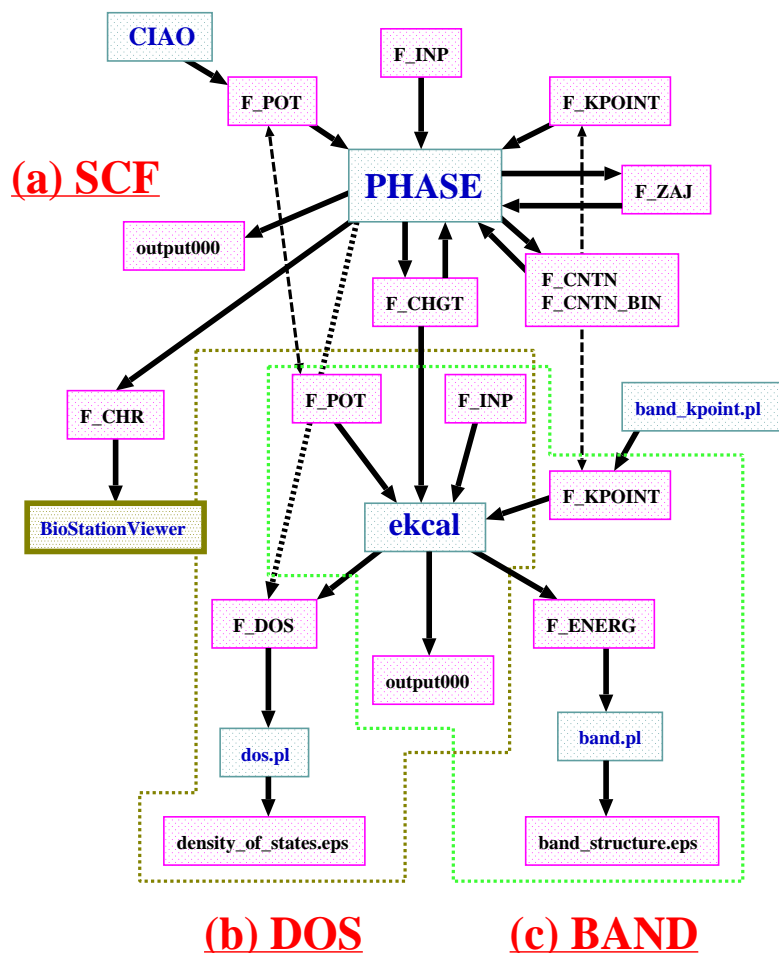


図 1: 処理過程のフローチャート. 実行ファイル PHASE に向かう矢印は入力を, 逆は出力を意味する. また, 青色と赤色はそれぞれ, 実行ファイルと入出力ファイルを指している. PHASE: 本プログラムの実行ファイル本体. 電荷密度分布の計算結果を F\_CHGT に出力. ekcal: F\_CHGT などを入力ファイルとして, 状態密度やバンド計算の計算結果を F\_DOS または F\_ENERG に出力. F\_INP: 入力ファイル. F\_POT: 擬ポテンシャル. F\_CHR: Gaussian cube ファイル形式による電荷密度分布の出力ファイル. band\_kpoint.pl:  $k$  点の入力データである F\_KPOINT を生成する Perl スクリプト. dos.pl: F\_DOS を入力ファイルとして, 状態密度の Postscript ファイル density\_of\_states.eps を生成する Perl スクリプト. band.pl: 各  $k$  点のエネルギーデータの出力ファイル F\_ENERG を基に, バンド構造を Postscript ファイル band\_structure.eps に出力する Perl スクリプト. output000: 計算結果の標準出力データ (ログファイル). Biostation viewer: 「革新的シミュレーションソフトウェアの研究開発」プロジェクトで開発された描画用ソフト.



表 1: 入出力ファイル一覧

ファイル ポインタ名	対応する実行 プログラム	入出力 の区別	デフォルト値	説明
F_INP	PHASE ekcal	入力	nfinp.data	結晶構造，計算精度，計算の制御などの情報が記述されています。
F_POT(n)	PHASE ekcal	入力	fort.37 等	擬ポテンシャルデータ。この配列に与えられた名前のファイルを読みこみます。各元素に 1 つのポテンシャルファイルが必要になります。
F_STOP	PHASE ekcal	入力	nfstop.data	F_INP で ifstop の値に正の整数が与えられている時に，途中で計算を停止させる回数をこのファイルから読み取ります。
F_KPOINT	PHASE ekcal	入力		サンプリング $k$ 点を与えるファイル。
F_DYNM	PHASE	出力	nfdynm.data	イオンが動く度に，その座標とイオンに働く力が出力されます。
F_ENF	PHASE	出力	nfefn.data	力を計算したときに，全エネルギーと力の最大値が出力されます。
F_CHR	PHASE	出力	nfchr.data	Gaussian cube ファイル形式による電荷密度分布の出力。
F_DOS	PHASE ekcal	出力	dos.data	状態密度の出力ファイル。
F_ENERG	ekcal	出力	nfenergy.data	ekcal で計算を行なうと，固有値が出力されます。
F_ZAJ	PHASE/ekcal	入出力	zaj.data	波動関数の情報が書き込まれる出力ファイルです。
F_CHGT	PHASE ekcal	入出力 出力	nfchgt.data	電荷密度の情報が出力されます。計算を継続する際には入力ファイルとなります。
F_CNTN	PHASE	入出力	continue.data	ジョブが途中終了した場合，継続するのに必要な情報を出力します。継続ジョブ投入の際には，このファイルを入力として使います。
F_CNTN _BIN	PHASE	入出力	continue _bin.data	同上。バイナリーファイル。
F_STATUS	PHASE/ekcal	出力	jobstatus00x	計算の進行状況が出力されます。

## 3 入力ファイル

### 3.1 F\_INP

PHASE を走らせるのに最低限必要なファイルは、擬ポテンシャルデータが格納されたファイルである F\_POT と、入力ファイルの F\_INP です。F\_INP は、ユーザーが PHASE にどのような計算をさせるのかを詳細に指定するために使われます。このファイルはタグ形式になっており、各タグに、結晶構造、計算精度、計算の制御などの情報を記入します。入力ファイルの記述方法は、次のようになります。

タグ形式	内部構造がない場合	識別子 = 入力値
ブロック形式	内部構造のある場合	識別子 { ... }
ブロック形式	複数の値を表形式で入力する場合	#tag col1 col2 col3
ブロック形式	デフォルト値の指定	#default col1 = val1, col2 = val2, col3 = val3 ...

この表の中で、col1, col2, col3 等は、それぞれ、コラム 1, コラム 2, コラム 3 などを表します。

第 3 の例では、ブロック内に #tag 識別子があると、表形式のブロックと解釈されることを示しています。第 4 の例が示すのは、#default 識別子により表形式ブロック内のコラムのデフォルト値を指定する方法です。

#default 識別子で指定したコラム名を、#tag 識別子で改めて指定する必要はありません。この表の各行で、#default 識別子で指定のあるコラムに入力した場合、行内での入力が優先されます。#default 識別子で指定された値を用いる場合には、\* を入力して下さい。ただし、最後のコラムの場合、あるいは同じ行の右側が全てデフォルト値を使用する場合には、\* を省略して空白にすることができます。

次に説明するのは、入力の際に使用可能な単位系です。その一覧を、以下の表に示します。ただし、デフォルトで使用される単位は太文字で表示されています。

長さ	<b>bohr</b> , angstrom, nm
エネルギー	<b>hartree</b> , eV, rydberg
時間	<b>au_time</b> , fs, ps, ns, s, sec, min, hour, day
速度	<b>bohr/au_time</b> , bohr/fs, angstrom/fs, angstrom/au_time, nm/fs, nm/au_time
力	<b>hartree/bohr</b> , hartree/angstrom, hartree/nm, eV/angstrom, eV/bohr, ev/nm, rydberg/bohr, rydberg/angstrom, rydberg/nm
圧力	<b>hartree/bohr<sup>3</sup></b> , hartree/angstrom <sup>3</sup> , hartree/nm <sup>3</sup> , eV/angstrom <sup>3</sup> , eV/bohr <sup>3</sup> , eV/nm <sup>3</sup> , rydberg/angstrom <sup>3</sup> , rydberg/bohr <sup>3</sup> , rydberg/nm <sup>3</sup> ,
質量	<b>au_mass</b> , atomic_mass,
角度	<b>degree</b> , radian
温度	<b>K</b> , centigrade

これらの単位を使うに当たって、大文字と小文字の区別はありません。また、指定された単位は、より下位のブロックに継承されます。ただし、速度や圧力のように単位に / が使われる場合、その前後にスペースを入れるとエラーになるので、ご注意ください。

他には、以下のような規則が入力仕様として定められています。

1. ブロック内の変数の全てに共通の単位を指定する場合は、ブロック中の先頭部分に

```
#units Bohr Hartree
```

等と記述します。

2. タグ形式で値が 1 つの実数値の場合、入力値を記入した後に空白を空けて、その後に単位を書くことができます。
3. 単位を省略した場合には、デフォルトとして原子単位が適用されます。
4. 入力値が 1 か 0 の整数の場合、yes, no や on, off も等価に扱われます。
5. コメント文を記入する場合には、! または // でコメント文を始めます。
6. 上記の仕様の例外として、!# は # と等価なものと思なされます。
7. 複数の変数に入力する場合、個々の入力式を改行して終らせる代わりに、"," でタグ形式文を一行に並べることができます。

8. 表形式のブロックは同じ名前のブロックタグを使って複数のブロックに分けて書くことができます。この場合、`#group` という識別子を使った、`#group id_str` という形の表記が必要になります。行を分割した場合には、異なる行には異なる `id_str` を入力して下さい。列を分割した場合には、同じ `id_str` を用いて下さい。
9. 二重引用符 “ ” で囲んだ文節は、間にスペースが入っても一つの文字列とみなされます。

次に、最も典型的な半導体である Si を題材に取り上げて、その入力ファイルの例を以下に示します。

```
Control{
  condition = initial    ! {0|1|2|3}
    ! {initial|continuation|fixed_charge|fixed_charge_continuation}
  cpumax = 1 day          ! {sec|min|hour|day}
  max_iteration = 6000
  nfstopcheck = 1
}

accuracy{
  cutoff_wf = 16.00 rydberg
  cutoff_cd = 64.00 rydberg
  num_bands = 20
  ksampling{
    method = mesh ! {mesh|file|direct_in|gamma}
    mesh{ nx = 4, ny = 4, nz = 4 }
    kshift{ k1 = 0.5, k2 = 0.5, k3 = 0.5 }
!
!           kpoints{
!           #tag kx ky kz denom weight
!           0 0 0 8 1
!           4 0 0 8 3
!           0 0 4 8 1
!           4 0 4 8 3
!           }
  }
  smearing{
    method = parabolic ! {parabolic|tetrahedral}
    width = 0.001 hartree
  }
  xctype = ldapw91      ! {ggapbe|ldapw91}
  scf_convergence{
    delta_total_energy = 1.e-12 hartree
    succession = 3      !default value = 3
  }
  force_convergence{
    max_force = 1.0e-4 ! (hartree/bohr)
  }
  ek_convergence{
    num_max_iteration = 300
    sw_eval_eig_diff = on
    delta_eigenvalue = 1.e-15 rydberg
    succession = 2
  }
  initial_wavefunctions = matrix_diagon ! {random_numbers|matrix_diagon}
  matrix_diagon{
    cutoff_wf = 16.00 rydberg
  }
}
```

```

}

structure{
  unit_cell_type = primitive
  unit_cell{
    #units bohr
    !a = 10.26000, b = 10.26000, c = 10.26000,
    !alpha=90.0, beta=90.0, gamma=90.0
    a_vector = 0.00000 5.13000 5.13000
    b_vector = 5.13000 0.00000 5.13000
    c_vector = 5.13000 5.13000 0.00000
  }
  symmetry{
    crystal_structure = diamond
    !tspace{
      ! lattice_system = f    !{t(rigonal|h|s|f|b|base)}
      ! num_generators = 3
      ! generators{
      !   #tag rotation  tx   ty   tz
      !       IE        0    0    0
      !       C3i+      0    0    0
      !       C4X+     1/4  1/2  3/4
      !   }
      !}
    sw_inversion = on
  }

  magnetic_state = para

  atom_list{
    coordinate_system = internal ! {cartesian|internal}
    atoms{
      #tag rx      ry      rz      weight  element  mobile
      0.125  0.125  0.125      2        Si      1
    }
  }
  element_list{
    #tag element  atomic_number  mass      zeta  deviation
    Si          14      51577.50  0.00  3.5
  }
}

wavefunction_solver{
  solvers{
    #tag sol      till_n  dts  dte  itr  var  prec  cmix
    MSD      5      0.2  1.0  40  linear  on  3
    lm+MSD   18      0.2  *    *    *    on  3
    RMM3     251     1.0  *    *    *    on  1
    MSD      -1      0.2  1.00  60  tanh   *  2
  }

  line_minimization{
    dt_lower_critical = 1.0
    dt_upper_critical = 5.0
  }
}

```

```

rmm{
    imGSrmm = 1
    rr_Critical_Value = 1.e-15
    edelta_change_to_rmm = 1.0e-4
}

subspace_rotation{
    subspace_matrix_size = 30
    damping_factor = 1.00
}
}

charge_mixing{
    num_mixing_methods = 3
    mixing_methods{
#tag no   method   rmxs   rmxe   itr   var   prec istr   nbmix   update
        1   simple   0.90   0.95   40   linear on
        2   simple   0.50   0.90  400   tanh   on
        3   broyden2 0.50   *      *      *      on    3    5    RENEW
    }
    charge_preconditioning{
        amix = 0.90
        bmix = -1.00
    }
}

structure_evolution{
    method = gdiis      !{quench|gdiis|cg|velocity_verlet}
    dt      = 100.0
    gdiis{
        gdiis_box_size = 4
        gdiis_hownew = RENEW
        c_forc2gdiis = 0.005d0
        c_iteration2GDIIS = 3
    }
}

postprocessing{
    dos{
        method = tetrahedral !{ tetrahedral|Gaussian }
        sw_dos      = ON
        deltaE_dos   = 1.e-4 hartree
        nwd_dos_window_width = 10
    }
    charge{
        sw_charge_rspace = ON
        filetype          = cube !{cube|density_only}
        title = "This is a title line for the bulk Si8"
    }
    STM{
        sw_stm_images = 0
        destm = 1 eV
    }
}

printoutlevel{

```

```

base = 1
timing = 1
solver = 1
evdfff = 0
rmm     = 0
snl     = 0
gdiis   = 1
eigenvalue = 1
spg     = 1
kp      = 2
pulay   = 0
matdiagon = 0
vlhxcq = 0
totalcharge = 1
submat  = 0
strcfctr = 0
}

```

### 3.2 識別子の説明

第 3.1 節で使われている各タグの定義を、表 2 に列記します。ただし、この表の中で、タグ名は太字で示されています。

表 2: 入力ファイル F\_INP に使用される識別子とその説明

第 1 ブロック 識別子名	第 2, 第 3 ブロック 識別子名	タグ識別子名	説明
<b>Control</b>			計算制御用ブロック識別子
		condition	preparation, -2: 入力座標の表示, 対称操作の生成, k 点の生成までで終了します。 automatic, -1: 継続可能であれば, 計算継続になります。そうでなければ, 計算開始になります。 initial, 0 : 計算開始 continuation, 1: 計算継続 (以下の 2 つは ekcal による計算で使用) fixed_charge, 2: 電荷を固定して計算 fixed_charge_continuation, 3: 固定電荷+計算継続 既定値は automatic です。
		cpumax	CPU 時間の上限 (デフォルト: 86400 sec) 単位: {sec, min, hour, day}
		max_iteration	繰り返し計算の上限 (デフォルト: 10000)
		nfstopcheck	ファイル nfstop.data に書かれた数値で, 処理を停止するべき更新回数を決定 (デフォルト: 1)
		sw_ekzaj	PHASE で, EKCAL で読み込める形式で F_ZAJ への出力を行うとき ON にする。EKCAL でそのファイルを読み込むときも ON にする。ただし, $\Gamma$ 点の計算でしか使用できません。 既定値は OFF です。
<b>accuracy</b>			計算精度の制御用ブロック識別子
		cutoff_wf	波動関数のカットオフエネルギー

表 2: 入力ファイル F\_INP に使用される識別子とその説明 (続き)

第 1 ブロック 識別子名	第 2, 第 3 ブロック 識別子名	タグ識別子名	説明
		cutoff_cd	電荷密度のカットオフエネルギー
		num_bands	バンド数
	<b>ksampling</b>		
		method	$k$ 点のサンプリング法。method の指定が無ければ、デフォルトの Monkhorst-Pack 法で $k$ 点を生成 monk: デフォルトの Monkhorst-Pack 法を明示的に指定する場合に使用します。 mesh: メッシュを生成します。 file: ファイルから入力 direct_in: 直接記述 gamma: $\Gamma$ 点のみ
	<b>mesh</b>		メッシュの分割数
		nx, ny, nz	x,y,z 方向への分割数 デフォルト値 = (4,4,4), 上限値 = (20,20,20)
	<b>kshift</b>		デフォルトの Monkhorst-Pack 法でのみ有効なタグ
		k1, k2, k3	メッシュのずれの指定 (入力値は [0.0, 0.5] の範囲) デフォルト値: hexagonal の場合: $k1 = k2 = 0, k3 = 0.5$ それ以外の場合: $k1 = k2 = k3 = 0.5$ ただし 0.5 はメッシュの刻み幅の半分の値を指す
	<b>kpoints</b>		$k$ 点の重みづけ
		kx ky kz denom weight	$\vec{k} = (kx/denom, ky/denom, kz/denom)$ $k$ 点の座標値と, その重みづけ
	<b>smearing</b>		$k$ 点サンプリングの smearing
		method	parabolic: Parabolic 法 (デフォルト) cold: Cold smearing 法 (金属系で有効) tetrahedron: Tetrahedron 法 improved_tetrahedron: 改良 tetrahedron 法 tetrahedron または improved_tetrahedron としたときには $k$ 点のサンプリングをメッシュ法にしなければなりません。
		width	smearing 幅 (デフォルト値: 0.001 hartree) method = parabolic と cold の時に使用
	(タグなし)		
		xctype	交換相関エネルギー (LDA, GGA) LDA: LDAPW91, PZ GGA: GGAPBE, REVPBE (デフォルト値に関しては, 第 22 節を参照)
	<b>scf_convergence</b>		自己無撞着場の収束判定
		delta_total_energy	原子あたりの全エネルギーの計算誤差の上限 $\Delta E$ (デフォルト値: $10^{-10}$ hartree)
		succession	誤差が $\Delta E$ 未満に何回連続して収まった時に計算を停止させるかを指定する回数 (デフォルト値: 3)
	<b>force_convergence</b>		力の収束判定

表 2: 入力ファイル F\_INP に使用される識別子とその説明 (続き)

第 1 ブロック 識別子名	第 2, 第 3 ブロック 識別子名	タグ識別子名	説明
		max_force	全原子に働く力の最大値がこの値より小さくなれば計算を停止 (デフォルト値: 0.001 hartree/bohr)
	ek_convergence		固有値の収束判定。ekcal による計算専用の識別子
		num_extra_bands	未収束でもかまわない追加のバンドの数 (デフォルト値: 2)
		num_max_iteration	$k$ 点一個当たりの最大の更新回数 (デフォルト値: 300)
		sw_eval_eig_diff	固有値評価用スイッチ
			{ 1, on, yes }: 評価あり (デフォルト) { 0, off, no }: 評価なし
		delta_eigenvalue	固有値の許容誤差 (デフォルト値: $10^{-15}$ hartree)
		succession	計算の繰り返し回数 (デフォルト値: 3)
	(タグなし)		
		initial_wavefunctions	波動関数の初期値 選択肢: {random_numbers, matrix_diagonal, atomic_orbitals, file} random_numbers: 乱数で初期化 matrix_diagonal: 小行列対角化で初期化 atomic_orbitals: 原子軌道で初期化 file: ファイル F_ZAJ から入力
	matrix_diagonal		波動関数の初期値を小行列対角化法で与える
		cutoff_wf	波動関数のカットオフ
	(タグなし)		
		initial_charge_density	電荷密度の初期値 選択肢: {Gauss, atomic_charge_density, file} Gauss: ガウス分布関数の重ね合わせで初期化 atomic_charge_density: 原子の電子密度の重ね合わせで初期化 file: ファイル F_CHGT から入力
	precalculation		
		nel_Ylm	予め計算してメモリー上に保持しておく球面調和関数の最高次数 (デフォルト値は 9)
structure			構造設定用ブロック識別子
	unit_cell_type		単位胞の型。選択肢: {primitive, Bravais}
	unit_cell		単位胞の指定。以下のいずれかの方法で与える
		a_vector b_vector c_vector	各格子ベクトルの $(x, y, z)$ 成分 デフォルトの単位は Bohr
		a, b, c alpha, beta, gamma	格子定数 a, b, c b-c 軸, c-a 軸, a-b 軸のなす角 (角度のデフォルト単位は degree)
	symmetry		
		method	選択肢: {manual, automatic} automatic を選択すると自動的に対称性を決定します。
		crystal_structure	選択肢:



表 2: 入力ファイル F\_INP に使用される識別子とその説明 (続き)

第 1 ブロック 識別子名	第 2, 第 3 ブロック 識別子名	タグ識別子名	説明
			{diamond, hexagonal, fcc, bcc, simple_cubic}
	<b>tspace</b>		柳瀬 章「空間群のプログラム TSPACE」(裳華房) および, ABCAP のマニュアルを参照
		lattice_system	{rhombohedral, trigonal, r, t, -1} {hexagonal, h, 0}, {primitive, simple, p, s, 1} {facecentered, f, 2}, {bodycentered, b, 3} {bottomcentered, basecentered, onefacecentered, bot, ba, o, 4}
		num_generators	生成元の数 (1 ~ 3 の整数値)
		generators	生成元
		af_generator	磁性空間群の生成元
	(第 3 タグなし)		
		sw_inversion	反転対称性の有無
	(第 2 タグなし)		
		magnetic_state	入力値: {para, antiferro, ferro} から選択 antiferro は af と省略も可
	<b>atom_list</b>		原子構成
		coordinate_system	選択肢: {cartesian, internal}
	<b>atoms</b>		
		rx, ry, rz	座標
		element	元素名
		mobile	可動性 入力値は {1, 0}, {on, off}, {yes, no} のどれでも可
		weight	重みづけ weight = 2 は sw_inversion = on の時のみ有効 このとき, 反転対称の位置にも原子を生成
	<b>element_list</b>		
		element	元素名 (atoms の element の入力値と一致させる)
		atomic_number	原子番号
		mass	質量
		zeta	スピン分極: $\zeta = (n_{\uparrow} - n_{\downarrow}) / (n_{\uparrow} + n_{\downarrow})$
		deviation	初期電荷をガウス関数の和で与えるときの各ガウス関数の偏差。 $\rho(r) = A \exp(-r^2/2\sigma^2)$ の $\sigma$ のように原子の電子密度分布の広がりを決めるパラメータ。 識別子名には dev や standard_deviation も使用可
<b>wavefunction_solver</b>	<b>solvers</b>		波動関数ソルバー 詳しくは第 8.3.1 節を参照
		sol	ソルバーの種類 MatrixDiagon: 行列対角化法 lm+MSD: lm(一次元探索) + MSD(改良型最急降下法) RMM2P, RMM3: RMM 法 MSD: 修正最速降下法 submat: subspace rotation (部分空間回転法)

表 2: 入力ファイル F\_INP に使用される識別子とその説明 (続き)

第 1 ブロック 識別子名	第 2, 第 3 ブロック 識別子名	タグ識別子名	説明
			Davidson: Davidson 法
		till_n	何回の更新まで, sol で指定された波動関数の更新 方法を適用するかを指定
		dt	計算開始時の時間刻み幅
		dte	itr で指定された更新の回数における時間刻み 幅。 dt の値のみが入力された場合には dte にも同 じ 値を適用。
		itr	時間刻み幅を変化させる回数の指定
		var	補間の形式。選択肢: {linear, tanh}
		prec	前処理の有無。選択肢: {on, off}
		cmix	電荷混合法の指定用変数。charge_mixing タグ の mixing_methods で指定されている, 各方法に 割り 振られた番号を使って指定する。
		submat	on のとき subspace_rotation の指定に従って subspace rotation を行う。選択肢: {on, off}
	<b>line_minimization</b>		一次元探索に関する制御
		dt_lower_critical	一次元探索をおこなう時の時間刻みの下限と上 限 (デフォルト値はそれぞれ, 0.005 と 2.0)
		dt_upper_critical	
		delta_lmdenom	
	<b>rmm</b>		残差最小化法
		imGSrmm	RMM 法で更新した波動関数に対して, Gram- Schmidt の直交化法を適用する頻度 (デフォルト値は, 毎回実行の imGSrmm = 1)
		rr_Critical_Value	バンド毎の収束判定条件。波動関数の残差のノ ルムがここで指定された値以下になれば, その バンドはそれ以降更新されない
		edelta_change	波動関数のソルバーを RMM 法に変えるとき の, _to_rmm 全エネルギー収束判定条件。ここで指定する値 より全エネルギーの収束が悪いときは, その前 のソルバーを続けて使う
	<b>subspace_rotation</b>		subspace 対角化に関する制御
		subspace_matrix_size	デフォルトの入力値はバンドの数 (num_bands) num_bands よりも大きな値が入力された場合 には, 強制的に num_bands の値を入力値に設 定。
		damping_factor	非対角要素のダンピング係数。[0.0, 1.0] の範囲 外の値が入力された場合には, 入力値を強制的 に 1.0 に設定。
		period	solver タグの submat が ON になっている場合, period に 1 回 subspace_rotation を行います。 例えば period=3 のとき iteration(i) のうち, i=1,4,7,10,... が subspace rotation を行う対象 になります。デフォルト値は 1。

表 2: 入力ファイル F\_INP に使用される識別子とその説明 (続き)

第 1 ブロック 識別子名	第 2, 第 3 ブロック 識別子名	タグ識別子名	説明
		critical_ratio	非対角項の要素の値 (1 要素あたり) と対角項の要素の値 (1 要素あたり) の比がいったん critical_ratio より小さくなった点に対しては、それ以後 subspace rotation を行いません。 デフォルト値は $10^{-15}$ 。
charge_mixing			電荷混合。詳しくは第 8.3.2 節を参照
	mixing_methods		電荷密度の混合法。デフォルトは simple_mixing
		method	選択肢: { simple, broyden2, pulay }
		rmxs	計算開始時の電荷密度を混ぜる割合 デフォルト値は 0.5
		rmxe	itr 回の更新の後に電荷密度を混ぜる割合。 デフォルト値は 0.5。rmxs の値のみが入力された場合には、rmxe にも同じ値を適用。
		itr	電荷密度の混合比 (rmx) を変化させる回数
		var	rmx を変化させる方法。選択肢: { linear, tanh }
		prec	前処理の有無。選択肢: { on, off }
		istr	method が simple 以外の場合に、istr 回の更新後に、指定した方法で電荷を混ぜる
		nbmix	蓄えておくべき電荷密度データの回数を指定
		update	nbmix 回分用意されている電荷密度の配列を使い切った時の処理の選択肢。 選択肢: { anew, renew } anew はそれまでのデータを全て棄却して新規に開始。renew は最も古いデータを最新のデータと入れ換える。
	charge_preconditioning		
		amix	前処理変数 a
		bmix	前処理変数 b
			(上記 2 変数の説明については第 8.3.2 節を参照)
structure_evolution			構造緩和計算用ブロック識別子
		method	選択肢: { sd, quench, gdiis, bfgs, cg, velocity_verlet }
		dt	時間刻み幅
	stress		ストレス計算
		sw_stress	ストレス計算の有無。選択肢: { on, off }
	gdiis		(GDIIS および BFGS を選択する場合のタグ)
		initial_method	GDIIS (BFGS) へ移行する前に利用する最適化アルゴリズム。選択肢: { quench, cg, sd }
		gdiis_box_size	ここで指定するイオン座標更新回数分のデータを gdiis(bfgs) 用配列に蓄える
		gdiis_hownew	gdiis_box_size で指定した回数分のイオン座標のデータ配列を使い切った時の処理法の選択 選択肢: { anew, renew }
		c_forc2gdiis	GDIIS (BFGS) への切替えの判定条件 デフォルト値は 0.0025 (hartree/bohr)

表 2: 入力ファイル F\_INP に使用される識別子とその説明 (続き)

第 1 ブロック 識別子名	第 2, 第 3 ブロック 識別子名	タグ識別子名	説明
<b>postprocessing</b>			
	<b>dos</b>		状態密度の出力
		sw_dos	状態密度出力の有無。選択肢: { on, off }
		method	選択肢: { tetrahedral, Gaussian }
		deltaE_dos	状態密度出力のエネルギー精度
		variance	method が Gaussian の場合のガウス関数の分散
		nwd_dos_window_width	出力時のエネルギー幅 $\Delta E$ を次式で指定: $\Delta E = \text{nwd\_window\_width} \times \text{deltaE\_dos}$
	<b>charge</b>		電荷の出力
		sw_charge_rspace	電荷出力の有無。選択肢: { on, off }
		filetype	電荷出力ファイルの形式 選択肢: { cube, density_only }
		title	電荷の出力ファイルの見出し filetype = cube の時のみ有効
<b>printoutlevel</b>			標準出力への出力レベルの制御 0: 出力なし 1: 情報を出力 2: デバッグ用の情報を出力
		base	他の変数に入力値が指定されていない時は、 この値がデフォルト
		pulay	Pulay 電荷混合法
		timing	時間指定情報
		solver	電子状態解法
		evdff	固有エネルギー差
		rmm	残差最小化法
		snl	非局所ポテンシャル
		gdiis	GDIIS 法
		eigenvalue	固有値
		spg	空間群
		kp	k 点
		matdiagon	行列対角法
		vlhxcq	ローカルポテンシャル
		totalcharge	電子密度
		submat	部分空間回転法
		strcfctr	構造因子
		parallel	並列化のための前処理の結果の出力制御
		input_file	入力ファイル F_INP の解析結果の出力
		parallel_debug	1 に設定するとゼロ番ノード以外のプロセスからも output00x_xxx といったファイルに出力を行う。
		jobstatus	計算の進行状況を jobstatus00x に出力
	<b>jobstatus_option</b>		状況ファイルの出力制御
		jobstatus_format	tag, tag_line, table が選択可能。既定値は tag です。
		jobstatus_series	ON または OFF

### 3.3 F\_POT(n)

F\_POT(n) で指定されるファイルには、第  $n$  番目の原子種に対する擬ポテンシャルデータが格納されている必要があります。その擬ポテンシャルデータは、本プログラム PHASE に付随して配布されているプログラム、CIAO で作成できます。

擬ポテンシャルを格納したファイルの最初の複数の連続した行には、# で始まるコメント文を記入することができます。もしコメント文を書き入れると、PHASE を走らせたときに、標準出力 (output000) に、そのコメント文が出力されます。

プログラム PHASE に擬ポテンシャルデータを読み込ませるには、その最初の 4 行 (コメント文がある場合には、コメント文以降の 4 行目まで) に、以下のパラメーターの値が指定されている必要があります。

1. natomn, ival, iloc, itpcc, igncpp

これらの変数は、それぞれ、原子番号  $Z$ 、価電子の数  $Z_v$ 、局在軌道の方位量子数  $l_{loc}$  に 1 を加えた値、コアチャージ補正の有 (≠1) 無 (=0)、擬ポテンシャルデータの形式 GNCPP1(=1), GNCPP2(=2) の指定に使われます。

2. xctype

交換相関相互エネルギーの型を指定します。選択できるのは、LDAPW91, GGAPBE の何れかです。

3. alp1, alp2, cc1, cc2

これらのパラメーターを  $\alpha_1, \alpha_2, c_1, c_2$  と書くと、PHASE の中では、コア部分の擬ポテンシャルを

$$V_{core} = -\frac{Z_v}{r} \{c_1 \text{erf}(\sqrt{\alpha_1} r) + c_2 \text{erf}(\sqrt{\alpha_2} r)\} \quad (1)$$

という式で近似して計算します。ただし、 $\text{erf}(\cdot)$  はガウスの誤差関数です。また、2 つの係数  $c_1$  と  $c_2$  の間には  $c_1 + c_2 = 1$  の関係があります。

4. nmesh, xh, rmax

動径方向のメッシュを

$$r_i = r_{max} \exp((i - N_{mesh})/x_h) \quad (i = 1, \dots, N_{mesh}) \quad (2)$$

の式にしたがって生成します。ただし、 $N_{mesh}$  は動径方向のメッシュの数を表します。

例として、第 4 節で例題として扱う Si 原子の擬ポテンシャルの最初の部分を以下に示します。

```
14  4  3  0  2 : zatom, ival, iloc, itpcc
ldapw91 : name
      2.160000  0.860000  1.605400  -0.605400 : alp,cc
      1501  96.000000  60.000000 : nmesh, xh, rmax
VALL
-0.14250064037552332E+07 -0.14102392478975291E+07 -0.13956251181755565E+07
-0.13811624288404209E+07 -0.13668496105922471E+07 -0.13526851103651347E+07
-0.13386673911985729E+07 -0.13247949320589846E+07 -0.13110662276746516E+07
-0.12974797883723934E+07 -0.12840341399159116E+07 -0.12707278233458301E+07
-0.12575593948213934E+07 -0.12445274254637859E+07 -0.12316305012010917E+07
-0.12188672226148657E+07 -0.12062362047882713E+07 -0.11937360771558125E+07
-0.11813654833546225E+07 -0.11691230810772763E+07 -0.11570075419261454E+07
-0.11450175512692606E+07 -0.11331518080976552E+07 -0.11214090248841981E+07
-0.11097879274438950E+07 -0.10982872547956155E+07 -0.10869057590252746E+07
-0.10756422051504281E+07 -0.10644953709862572E+07 -0.10534640470129563E+07
-0.10425470362444966E+07 -0.10317431540987322E+07 -0.10210512282688706E+07
-0.10104700985962711E+07 -0.99999861694454885E+06 -0.98963564707499891E+06
.....
.....
.....
```

1 行目から 4 行目までは、すでに説明した通りです。価電子 4 個を持つ原子番号 14 の Si 原子の、LDAPW91 法による擬ポテンシャルであることが、これらの行から分かります。

5 行目 (コメント文がある場合には 6 行目) に書かれている VALL というのは, PHASE のプログラム内で擬ポテンシャルのチェック用に使われる記号です。

その次の行からが擬ポテンシャルの実際のデータです。このデータの最初のブロックは, 遮蔽された全電子ポテンシャル (screened All Electron potential,  $V_{scr}^{AE}(r)$ ) に関するもので, そのデータ形式は,

```
do ir = 1, nmesh
   $V_{scr}^{AE}(ir)$ 
end do
```

という形になっています。

第 2 のブロックは, 遮蔽された局所ポテンシャル (screened local potential,  $V_{scr,iloc}^{PP}(r,l)$ ) に関するものです。 $V_{scr}^{AE}(r)$  同様, そのデータ形式は,

```
do ir = 1, nmesh
   $V_{scr,iloc}^{PP}(ir,iloc)$ 
end do
```

となります。

第 3 のブロックは, 価電子の電荷密度 (valence charge density,  $n_v(r)$ ) に, 球面の面積  $4\pi r^2$  をかけたものです。これを  $\rho_v(r)$  とすると ( $\rho_v(r) = 4\pi r^2 n_v(r)$ ), そのデータは,

```
do ir = 1, nmesh
   $\rho_v(r)$ 
end do
```

と書かれています。

これらの 3 ブロックの記述が終わった後に, 軌道別に擬波動関数と擬ポテンシャルのデータが出力されます。その形式は, ノルム保存の場合とウルトラソフトの場合で全く異なります。詳しくは, CIAO のユーザーマニュアルをご参照ください。

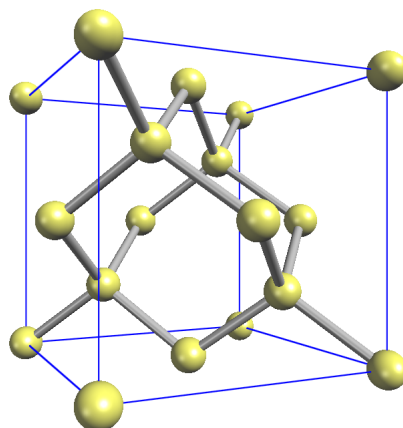


図 2: Si 結晶の構造図

## 4 計算例 1 : Si 結晶の全エネルギー計算 (Si<sub>8</sub>)

### 4.1 入力データ

PHASE の実行例として, Si 結晶 (図 2 参照) の全エネルギー計算について説明します。最初に, `file_names.data` というファイルについて説明します。このファイルは, 使用者が入出力ファイル名を指定する際に使われます。

ここでは, PHASE に同梱されているサンプル Si8 に沿って説明することになります。samples というディレクトリの中に, Si8 というディレクトリーがあります。この中にある `file_names.data` には,

```
&fnames
F_INP      = './input_scf_Si8.data'
F_POT(1)   = '../pp/Si_ldapw91_nc_01.pp'
F_DYNM     = './nfdynm.data'
F_CHGT     = './nfchgt.data'
F_ENF      = './nfefn.data'
F_ZAJ      = './zaj.data'
F_CHR      = './nfchr.cube'
&end
```

という記述があります。この第一行目で指定されている `input_scf_Si8.data` というファイルには, 原子の座標などを記述した入力データが書かれています。

ファイル `file_names.data` の 3 行目 (あるいは `fnames` リストの 2 行目) の, `F_POT(1)` で指定されている `Si_ldapw91_nc_01.pp` というファイルには, LDA-PW91 の交換相関汎関数を用いて計算された, Si の擬ポテンシャルが格納されています。他の元素に関する擬ポテンシャルも, PHASE に付属の別プログラム CIAO で計算することができます。

次に, 入力ファイル `F_INP` の説明に移りますが, その詳細は, 表 2 ですすでに説明しました。ここでは, 入力ファイルの構成の概略を, 表に簡単にまとめておきます。

第 1 タグ名	役割	関連箇所
Control	新たに計算を開始するのか，それとも過去に行なった計算結果を基に，その続きを行なうのかの別，あるいは計算に掛かる最大許容時間の設定などに使います。	第 4.5 節
accuracy	収束条件の設定などを行ないます。	第 4.6 節 ～ 第 4.8 節
structure	原子構造や格子定数などの初期値の入力に使います。	第 4.9 節 ～ 第 4.12 節 第 6.1 節
wavefunction_solver	波動関数のソルバーの選択や，それらの切替えを行なうタイミングなどを指定します。	第 8.3.1 節
charge_mixing	電荷混合法と，計算結果の更新の前後での電荷混合の比率とを選択します。	第 8.3.2 節
structure_evolution	原子構造の緩和過程の計算に使用します。	第 8.5 節
postprocessing	後処理の過程を指定するタグ。	第 6.5 節

入力ファイルの例として，ここでは，samples に含まれているサンプルのひとつである Si8 の入力ファイル input\_scf.Si8.data について説明します。

このファイルには，以下のような入力データが記入されています。

```
Control{
    cpumax = 3600 sec      ! {sec|min|hour|day}
}

accuracy{
    cutoff_wf = 9.00 rydberg
    cutoff_cd = 36.00 rydberg
    num_bands = 20
    ksampling{
        method = mesh ! {mesh|file|direct_in|gamma}
        mesh{ nx = 4, ny = 4, nz = 4 }
    }
    smearing{
        method = tetrahedral ! {parabolic|tetrahedral}
        width = 0.001 hartree
    }
    xctype = ldapw91
    scf_convergence{
        delta_total_energy = 1.e-12 hartree
        succession = 3 !default value = 3
    }
    force_convergence{
        max_force = 0.1e-3
    }
    initial_wavefunctions = matrix_diagon
        !{random_numbers|matrix_diagon|atomic_orbitals}
    matrix_diagon{
        cutoff_wf = 3.00 rydberg
    }
    initial_charge_density = Gauss
        !{Gauss|atomic_charge_density}
}

structure{
    unit_cell{
        a_vector = 10.26 0.00 0.00
```



```

        b_vector = 0.00 10.26 0.00
        c_vector = 0.00 0.00 10.26
    }
    atom_list{
        coordinate_system = internal ! {cartesian|internal}
        atoms{
            #default weight = 1, element = Si, mobile = 1
            #tag    rx        ry        rz
                0.125    0.125    0.125
            -0.125    -0.125    -0.125
                0.125    0.625    0.625
            -0.125    -0.625    -0.625
                0.625    0.125    0.625
            -0.625    -0.125    -0.625
                0.625    0.625    0.125
            -0.625    -0.625    -0.125
        }
    }
    element_list{
        #tag element  atomicnumber
            Si        14
    }
}

wavefunction_solver{
    solvers{
        #tag    sol        till_n    dts    dte    itr    var    prec    cmix    submat
            lm+MSD    10        0.2    *        *        *        *        on    1    on
            RMM3      -1        1.0    *        *        *        *        on    1    on
    }
}

charge_mixing{
    mixing_methods{
        #tag no    method    rmxs    rmxe    itr    var    prec
            1    simple    0.90    0.95    40    linear    on
    }
}

Postprocessing{
    dos{
        method = tetrahedral
        sw_dos = ON
        deltaE      = 1.e-4 hartree
        deviation_Gaussian = 1.e-5
        nwd_dos_window_width = 10
    }
    charge{
        sw_charge_rspace = ON
        filetype          = cube  !{cube|density_only}
        title = "This is a title line for the bulk Si"
    }
}

```

この入力ファイルを用いれば、波動関数のカットオフエネルギーは 9.0 Ry、電荷密度分布のカットオフエネルギーは 36.0 Ry、単位胞は  $a \times a \times a$  ( $a = 10\text{Bohr}$ )、 $k$  点のメッシュは  $4 \times 4 \times 4$  で対称性なし ( $\text{sw\_inversion} = \text{off}$ )

という条件で計算されます。これは、第一象限を  $\Gamma$  点を含まない  $2 \times 2 \times 2$  メッシュに分割し、その分割を全ての象限に広げて  $4 \times 4 \times 4$  メッシュとして、時間反転対称を考慮して  $4 \times 4 \times 2$  のメッシュ点で計算することを意味します。

交換相関汎関数は `xctype = ldapw91` で指定されており、スピン分極はありません、計算するエネルギー準位数は `accuracy` というタグの中で `num_bands = 20` と指定されています。エネルギー準位数としては系の全電子数をスピン多重度で割った数の 2 割強程度が目安になります。ここでは、原子数が 8、原子あたりの電子数が 4 なので、系の全電子数は 32 でその半数は 16、さらにこの値の 2 割程度の整数を加えて 20 としました。

エネルギー準位の収束判定は、同じタグの中で、

```
scf_convergence{
    delta_total_energy = 1.e-12 hartree
    succession      = 3    !default value = 3
}
```

として指定されています。これは、計算誤差が  $\Delta E < 10^{-12}$  Hartree の範囲に 3 回続けて収まったら計算を終了させることを意味します。

同様に、力の計算の収束判定は、やはり同じタグの中で、

```
force_convergence{
    max_force = 0.1e-3
}
```

(単位は Hartree/Bohr) で指定されています。

原子構造に関しては、この例の場合、立方体の単位胞内に 8 個の Si 原子がダイヤモンド構造に配置した Si 完全結晶の電子状態を計算することができます。座標位置は単位胞内座標で指定されていますが、`coordinate.system = cartesian` とすれば、デカルト座標を使って Bohr 単位で入力することも可能です。

`atom_list` では、`weight`, `element`, `mobile` の値にデフォルト値が設定されていますので、各原子に個別に与えられたこれらの変数の値を省略することも可能です。このマニュアルで扱う他の入力ファイルの例では、デフォルト値が設定されている場合、個々の原子の入力値は省略します。

## 4.2 計算の実行と途中経過

これらのファイルを用意した後、MPI を使って PHASE を実行します。

```
% mpirun -np A ../bin/phase ne=B nk=C
```

とタイプして下さい。ここで、 $A, B, C$  はそれぞれ、計算に使用するプロセッサの数、エネルギー準位の分割計算の数、および、 $k$  点の分割計算の数を指します。これらのパラメーターの値の間には、 $A = B \times C$  という関係が成り立つ必要があります。

計算の途中経過は、ログファイル `output000` に出力されますので、

```
% grep TOTAL output000
```

として、全エネルギーの収束状況を確認することができます。

## 4.3 出力データ

計算が終了すると、次のファイルが出力されます。

```
continue.data
continue_bin.data
dos.data
nfchgt.data
nfefn.data
nfchr.data
nfdynm.data
output000
zaj.data
```

複数のプロセッサをもつマシンで並列計算を行なった場合には、ノードの数に応じて、output000\_001 等のファイルも出力されます。

先頭のノードから出力されたログファイルである output000 の、始めの部分には、

```
F_POT(1)  = ../pp/Si_ldapw91_nc_01.pp
F_INP     = ./input_scf_Si8.data
F_CHGT    = ./nfchgt.data
F_ZAJ     = ./zaj.data
F_DYNM    = ./nfdynm.data
F_ENF     = ./nfefn.data
```

のように、入出力ファイルなどの情報が書かれています。

波動関数を更新した各回毎に、全エネルギーと、その内訳が出力されています。この計算例の場合には、最後の更新が 27 回目、output000 の該当部分には、

```
edeltb = -0.3588D-12 edelta = 0.1000D-11 ntcnvg = 2
TOTAL ENERGY FOR 27 -TH ITER= -31.587937147369 edel = -0.117240D-12
KI= 11.9078509 HA= 2.1406508 XC= -9.5483626 LO= -10.2841067
NL= 7.7978897 EW= -33.6018591 PC= 0.0000000 EN= 0.0000000
! edeltb = -0.117240D-12 hr (= -0.146549D-13 hr/atom ) ( iter = 27 )
edeltb = -0.1172D-12 edelta = 0.1000D-11 ntcnvg = 3
```

という記述が見つかるはずです。ここで、KI は運動エネルギー、HA は Hartree エネルギー、LO は局所エネルギー、NL は非局所エネルギー、EW は Ewald エネルギー、PC はコア補正エネルギー、EN はエントロピーを表します。これらの総和が全エネルギーになります。

Fermi 準位等のエネルギー順位に関する情報も、ログファイルに出力されています。

```
EFermi = 0.24579615
===== Energy Eigen Values =====
1 0.00000000 0.00000000 0.00000000
-0.19655861 -0.04839227 -0.04839227 -0.04839227 -0.04839227
-0.04839227 -0.04839227 0.12584623 0.12584623 0.12584623
0.12584623 0.12584623 0.12584623 0.23389619 0.23389619
0.23389619 0.26196708 0.26196708 0.26196708 0.26196708
2 0.25000000 0.00000000 0.00000000
-0.18998394 -0.11270106 -0.04555873 -0.04555873 -0.04555873
-0.04555873 0.02675145 0.10512408 0.10512408 0.10512408
0.10512408 0.13505063 0.13505063 0.18575457 0.20251681
0.20251681 0.25769611 0.29275976 0.30811466 0.30811466
3 0.50000000 0.00000000 0.00000000
-0.16102016 -0.16102016 -0.04095243 -0.04095243 -0.04095243
-0.04095243 0.08874423 0.08874423 0.08874423 0.08874423
0.10781439 0.10781439 0.16184290 0.16184290 0.16184290
0.16184290 0.27543069 0.27543069 0.35154734 0.35154734
4 0.75000000 0.00000000 0.00000000
-0.18998394 -0.11270106 -0.04555873 -0.04555873 -0.04555873
-0.04555873 0.02675145 0.10512408 0.10512408 0.10512408
0.10512408 0.13505063 0.13505063 0.18575457 0.20251681
0.20251681 0.25769611 0.29275976 0.30811466 0.30811466
5 0.00000000 0.25000000 0.00000000
-0.18998394 -0.11270106 -0.04555873 -0.04555873 -0.04555873
-0.04555873 0.02675145 0.10512408 0.10512408 0.10512408
0.10512408 0.13505063 0.13505063 0.18575457 0.20251681
0.20251681 0.25769611 0.29275976 0.30811466 0.30811466
.....
.....
.....
```

次は占有数に関する情報です。

```

===== Occupations =====
  1      0.00000000      0.00000000      0.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      0.00000000      0.00000000      0.00000000      0.00000000
  2      0.25000000      0.00000000      0.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      0.00000000      0.00000000      0.00000000      0.00000000
  3      0.50000000      0.00000000      0.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      1.00000000      1.00000000      1.00000000      1.00000000
  1.00000000      0.00000000      0.00000000      0.00000000      0.00000000

```

占有数は 0~1 の間の値をとるので、例えば、スピンを考えない計算の場合、1.0 は電子 2 個に占有されることを表わすことに注意する必要があります。

MD の全エネルギーの経過は nfefn.data に出力されます。今の場合、

```

iter_ion, iter_total, etotal, forcmx
  1      27      -31.5879371474      0.0000000016

```

となっています。

#### 4.4 強制終了の方法

計算がなかなか収束しない場合に、途中経過をファイルに出力させて強制的に計算を正常終了させる方法が PHASE には用意されています。この強制終了を実際に行なうには、nfstop.data に、現在のループ回数より小さな値 (例えば 1) を書き込みます。

#### 4.5 再計算の方法

再計算を行う場合は、

```

continue_bin.data
continue.data
zaj.data
nfchgt.data

```

が存在することを確認してから、入力ファイルの Control のタグに、

```
condition = continuation
```

と書き込みます。condition を automatic にしておけば、再計算可能なときには、自動的に再計算します。この変数のデフォルトは automatic ですが、他にも以下のような選択肢があります。

-2	-1	0	1
preparation	automatic	initial	continuation
2	3		
fixed_charge	fixed_charge_continuation		

変数 condition への入力、この表の上段にある数字と、下段の文字列のどちらでも可能です。

## 4.6 交換相関エネルギーの型の指定方法

交換相関エネルギーの型の指定は, 入力ファイルにある `xctype` という変数に代入することで行ないます。選択できるのは, LDA としては LDAPW91 か PZ, GGA としては GGAPBE, REVPBE があります。

## 4.7 $k$ 点の指定方法

$k$  点のサンプリング法を指定しているのは, 入力ファイルの

```
accuracy{
  ksampling{
    method = mesh ! {monk|mesh|file|direct_in|gamma}
    mesh{  nx = 4, ny = 4, nz = 4  }
  }
}
```

の部分です。

```
mesh{  nx = 4, ny = 4, nz = 4  }
```

の部分で, Brillouin 領域の第一象限 (すなわち,  $0 < k_1 < \pi/a_1$ ,  $0 < k_2 < \pi/a_2$ ,  $0 < k_3 < \pi/a_3$ ) を  $4 \times 4 \times 4$  に等分割して, 各直方体の中心に  $k$  点位置を定めます。

## 4.8 Brillouin 領域の積分方法

入力ファイルの例では Brillouin 領域の積分方法として parabolic broadening 法と四面体法, cold smearing 法が使用できます。

```
accuracy{
  smearing{
    method = parabolic ! {parabolic|tetrahedral|cold}
    width  = 0.001 hartree
  }
}
```

## 4.9 2 種類以上の原子を扱う方法

第 4.1 節の入力ファイルの例では, Si 原子のみの 1 元素を扱っていますが, これを拡張して, 2 種類以上の原子を指定する方法を説明します。例えば, 第 1 番目の原子を Si 原子から Ge 原子に変えるには, 入力ファイルの `structure` タグの `atom_list` と `element_list` の部分を, 以下のように変更します。

```
atom_list{
  coordinate_system = internal ! {cartesian|internal}
  atoms{
    #default weight = 1, element = Si, mobile = 1
    #tag   rx       ry       rz       element
         0.125    0.125    0.125        Ge
        -0.125   -0.125   -0.125
         0.125    0.625    0.625
        -0.125   -0.625   -0.625
         0.625    0.125    0.625
        -0.625   -0.125   -0.625
         0.625    0.625    0.125
        -0.625   -0.625   -0.125
  }
}
```

```

element_list{ #tag element  atomicnumber
                Si           14
                Ge           32
            }

```

default で weight = 1, element = Si, mobile = 1 と指定されているので, element = Ge と入力されている以外の元素名と, 全ての原子の weight と mobile という変数には, デフォルト値が使われます。

これと同時に, file\_names.data に Ge のポテンシャルファイルを追記します。そのファイル名を Ge\_ldapw91\_nc\_01.pp とすると, file\_names.data は,

```

&fnames
  F_INP      = './input_scf_Si8.data'
  F_POT(1) = './pp/Si_ldapw91_nc_01.pp'
  ...
&end

```

から,

```

&fnames
  F_INP      = './input_Si.data'
  F_POT(1) = './pp/Si_ldapw91_nc_01.pp'
  F_POT(2) = './pp/Ge_ldapw91_nc_01.pp'
  ...
&end

```

に書き換えることになります。原子の種類が 3 種類以上でも, 同様の変更を施せば計算が可能です。ただし, 17 種類以上は指定できません。

#### 4.10 原子座標を原子単位系で入力する方法

本節の入力ファイルの例では, 原子位置を単位胞を基準にした結晶座標系で指定していますが, 原子単位系を用いて Bohr 単位で座標を入力する方法を説明します。このためには, 入力ファイルの

```

atom_list{
    coordinate_system = internal ! {cartesian|internal}
}

```

を

```

atom_list{
    coordinate_system = cartesian ! {cartesian|internal}
}

```

と変更すれば, デカルト座標を使って Bohr 単位で入力することも可能です。

#### 4.11 反転中心を考慮する方法

反転対称性を考慮して入力するには, 反転中心の原点を  $r = 0$  として, 入力ファイルの structure タグで

```

symmetry{
    sw_inversion = on
}

```

とします。詳しくは, 第 5.3 節で説明します。

#### 4.12 分子動力学で指定原子を固定する方法

入力ファイルの例では全ての Si 原子を動かすように指定されています。これに代わって、一部の原子を固定する方法を説明します。例えば、第 1 番～第 4 番までの Si 原子を固定して、残りの第 5 番～第 8 番までの Si 原子を動かすには、入力ファイルの `atom_list` の部分を次のように書き換えてやります。

```
atom_list{
  coordinate_system = internal ! {cartesian|internal}
  atoms{
    #default weight = 1, element = Si, mobile = 1
    #tag    rx      ry      rz      mobile
          0.125    0.125    0.125      0
        -0.125   -0.125   -0.125      0
          0.125    0.625    0.625      0
        -0.125   -0.625   -0.625      0
          0.625    0.125    0.625
        -0.625   -0.125   -0.625
          0.625    0.625    0.125
        -0.625   -0.625   -0.125
  }
}
```

これで、第 1 番～第 4 番までの Si 原子の可動性 (`mobile`) が失われて、原子位置が固定されます。

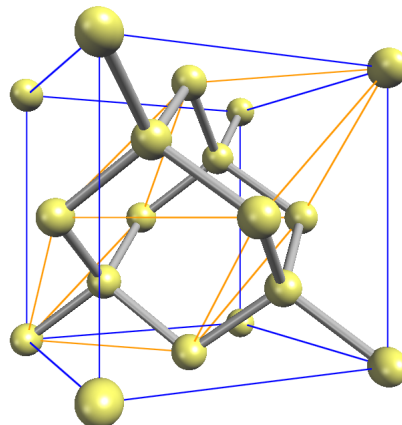


図 3:  $\text{Si}_2$  の原子構造図。黄線は基本格子，青線はブラベ格子を表す。

## 5 計算例 2：対称性を考慮した計算 ( $\text{Si}_2$ )

PHASE では、対称性を考慮することにより、ブリルアンゾーン積分に使われる標本点の数を減らすことができます。本節では、 $\text{Si}$  結晶 (図 3 参照) を例にとり、このやり方について説明します。

対称性に関する指定は、入力ファイル中の第 1 タグ `structure` (表 1 参照) で行ないます。以下では、`structure` 内で用いられる変数の指定の仕方について説明します。

### 5.1 単位胞の指定： $\text{Si}_2$ の場合

原子構造を指定する方法には、基本格子を指定する方法とブラベ格子を指定する方法とがあります。具体的には、変数 `unit_cell_type` の入力値を `primitive` か `Bravais` のどちらかから選択することで指定できます。

#### 5.1.1 単位胞を基本格子で指定

第 3.1 節にあるように、

```
unit_cell_type = primitive
unit_cell{
  #units bohr
  a_vector = 0.00000 5.13000 5.13000
  b_vector = 5.13000 0.00000 5.13000
  c_vector = 5.13000 5.13000 0.00000
}
```

と入力します。

#### 5.1.2 単位胞を格子定数で指定

この場合、指定の仕方は以下ようになります。

```
unit_cell_type = Bravais
unit_cell{
  #units bohr
  a = 10.26, b = 10.26, c = 10.26
  alpha = 90, beta = 90, gamma = 90
}
```



晶系	格子定数	unit_cell に記述する値	格子の種類	lattice.system に指定する単語
立方 (c)	$a$	$a=a, b=a, c=a$ $\alpha=90, \beta=90, \gamma=90$	単純 (P)	primitive
			面心 (F)	facecentered
			体心 (I)	bodycentered
正方 (t)	$a, c$	$a=a, b=a, c=c$ $\alpha=90, \beta=90, \gamma=90$	単純 (P)	primitive
			体心 (I)	bodycentered
直方 (o)	$a, b, c$	$a=a, b=b, c=c$ $\alpha=90, \beta=90, \gamma=90$	単純 (P)	primitive
			底心 (C)	basecentered
			面心 (F)	facecentered
			体心 (I)	bodycentered
六方 (h)	$a, c$	$a=a, b=a, c=c$ $\alpha=90, \beta=90, \gamma=120$	単純 (P)	hexagonal
三方 (h) 菱面体軸	$a, c$	$a=a, b=a, c=c$ $\alpha=90, \beta=90, \gamma=120$	菱面 (R)	rhombohedral
六方晶軸			単純 (P)	hexagonal
単斜 (m)	$a, b, c$ $\beta$	$a=a, b=b, c=c$ $\alpha=90, \beta=\beta, \gamma=90$	単純 (P)	primitive
			底心 (C)	basecentered
三斜 (a)	$a, b, c$ $\alpha, \beta, \gamma$	$a=a, b=b, c=c$ $\alpha=\alpha, \beta=\beta, \gamma=\gamma$	単純 (P)	primitive

表 3: ブラベー格子と晶系。

ブラベ格子を指定して入力した場合、次節で説明する対称性の指定により、プログラムの側で基本格子を決定します。菱面体晶系 (rhombohedral) の場合には、対応する六方晶系 (hexagonal) の格子定数を入力します。六方晶系と菱面体晶系の基本並進ベクトルの関係を図 4 に示します。また、原子座標を内部座標で入力する場合は、等価原子を除いて、単位胞内のすべて原子の位置を結晶軸ベクトル (慣用単位胞の 3 辺を表すベクトル) に対する相対座標 (ワイコフ位置の原子座標) で入力します。デカルト座標で入力する場合は、表 4 に示されている基本並進ベクトルと整合するように入力してください。

## 5.2 対称性の指定

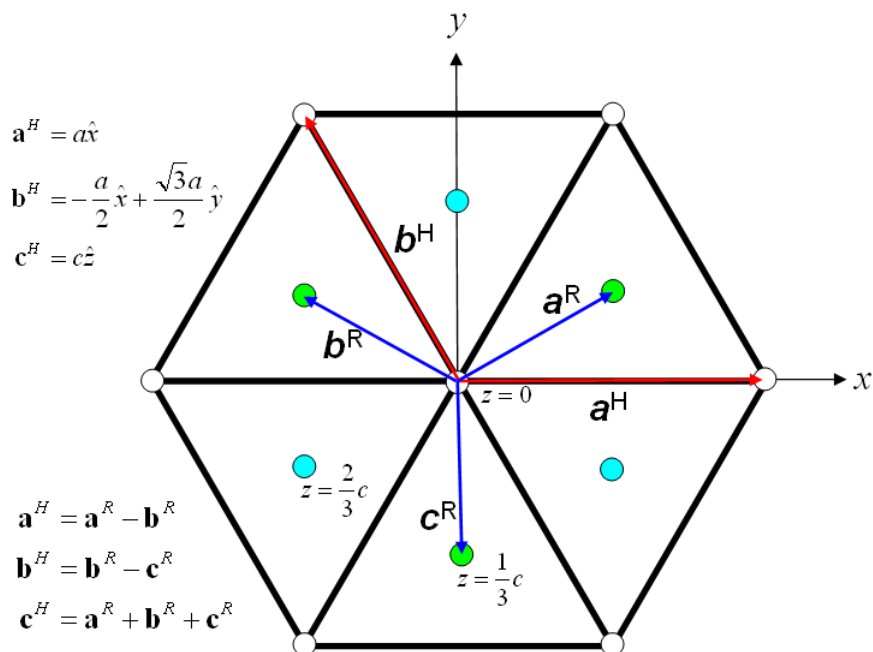
対称性の指定のやり方には、結晶構造を入力する方法、生成元を入力する方法、対称操作を自動的に決定する方法とがあります。

### 5.2.1 結晶構造を入力する方法

変数 `crystal.structure` に、結晶構造の型を入力します。この場合、選択肢には `diamond`, `hexagonal`, `fcc`, `bcc`, `simple_cubic` の 5 通りがあることは、表 2 で説明した通りです。Si 結晶の場合に選択すべき結晶構造は `diamond` です。

ブラベー格子	a	b	c
単純立方 (cP)	$a\hat{x}$	$a\hat{y}$	$a\hat{z}$
面心立方 (cF)	$\frac{a}{2}(\hat{y} + \hat{z})$	$\frac{a}{2}(\hat{x} + \hat{z})$	$\frac{a}{2}(\hat{x} + \hat{y})$
体心立方 (cI)	$\frac{a}{2}(-\hat{x} + \hat{y} + \hat{z})$	$\frac{a}{2}(\hat{x} - \hat{y} + \hat{z})$	$\frac{a}{2}(\hat{x} + \hat{y} - \hat{z})$
単純正方 (tP)	$a\hat{x}$	$a\hat{y}$	$c\hat{z}$
体心正方 (tI)	$\frac{1}{2}(-a\hat{x} + a\hat{y} + c\hat{z})$	$\frac{1}{2}(a\hat{x} - a\hat{y} + c\hat{z})$	$\frac{1}{2}(a\hat{x} + a\hat{y} - c\hat{z})$
単純直方 (oP)	$a\hat{x}$	$b\hat{y}$	$c\hat{z}$
底心直方 (oC)	$\frac{1}{2}(a\hat{x} - b\hat{y})$	$\frac{1}{2}(a\hat{x} + b\hat{y})$	$c\hat{z}$
面心直方 (oF)	$\frac{1}{2}(b\hat{y} + c\hat{z})$	$\frac{1}{2}(a\hat{x} + c\hat{z})$	$\frac{1}{2}(a\hat{x} + c\hat{y})$
体心直方 (oI)	$\frac{1}{2}(-a\hat{x} + b\hat{y} + c\hat{z})$	$\frac{1}{2}(a\hat{x} - b\hat{y} + c\hat{z})$	$\frac{1}{2}(a\hat{x} + b\hat{y} - c\hat{z})$
単純六方 (hP)	$a\hat{x}$	$a(-\frac{1}{2}\hat{x} + \frac{\sqrt{3}}{2}\hat{y})$	$c\hat{z}$
単純菱面体 (hR)	$\frac{a}{2}\hat{x} + \frac{a}{2\sqrt{3}}\hat{y} + \frac{1}{3}c\hat{z}$	$-\frac{a}{2}\hat{x} + \frac{a}{2\sqrt{3}}\hat{y} + \frac{1}{3}c\hat{z}$	$-\frac{a}{\sqrt{3}}\hat{y} + \frac{1}{3}c\hat{z}$
単純単斜 (mP)	$a\hat{x}$	$b\hat{y}$	$c(\cos\beta\hat{x} + \sin\beta\hat{z})$
底心単斜 (mC)	$\frac{1}{2}(a\hat{x} - b\hat{y})$	$\frac{1}{2}(a\hat{x} + b\hat{y})$	$c(\cos\beta\hat{x} + \sin\beta\hat{z})$
単純三斜 (aP)	$a\hat{x}$	$b(\cos\gamma\hat{x} + \sin\gamma\hat{y})$	$c\left(\cos\beta\hat{x} + \frac{\cos\alpha - \cos\beta\cos\gamma}{\sin\gamma}\hat{y} + \sqrt{1 - \frac{\cos^2\alpha + \cos^2\beta - 2\cos\alpha\cos\beta\cos\gamma}{\sin^2\gamma}}\hat{z}\right)$

表 4: ブラベー格子の基本並進ベクトル。

図 4: 六方晶系と菱面体晶系の関係。六方軸の方から見た格子点と基本並進ベクトルが示されています。 $a^H, b^H, c^H$  は六方晶系の基本並進ベクトルで,  $a^R, b^R, c^R$  は菱面体晶系の基本並進ベクトルです。

### 5.2.2 生成元を入力する方法

前述の `crystal_structure` という変数の代わりに、タグ `tSPACE` を用います。Si 結晶の場合、`tSPACE` の入力値は以下ようになります。

```
tSPACE{
  lattice_system = facecentered
  !{rhombohedral|hexagonal|primitive|facecentered|bodycentered|basecentered}
  num_generators = 3
  generators{
    #tag rotation  tx  ty  tz
      IE          0   0   0
      C31+        0   0   0
      C4X+       1/4  1/2  3/4
  }
}
```

予め、面心格子を使うことを `lattice_system = facecentered` で、また、生成元の数 3 であることを `num_generators = 3` で宣言した後で、タグ `generators` の中で、`IE`、`C31+`、`C4X+` が、具体的に 3 種類の生成元を指定しています。これらの生成元の詳細については、ABCAP マニュアルの第 5.1.1 節をご参照ください。

bcc 格子の反強磁性 Cr の場合、 $(1/2, 1/2, 1/2)$  並進にスピン反転が伴うして、`unit_cell` と `symmetry` の設定は以下ようになります。

```
unit_cell_type = Bravais
unit_cell{
  a = 5.65, b = 5.65, c = 5.65
  alpha = 90, beta = 90, gamma = 90
}
magnetic_state = af
symmetry{
  tSPACE{
    lattice_system = primitive
    generators{
      #tag rotation  tx  ty  tz
        5           0/1 0/1 0/1
        19          0/1 0/1 0/1
    }
    af_generator{
      #tag rotation  tx  ty  tz
        1           1/2 1/2 1/2
    }
  }
}
```

反強磁性状態の計算をするために、`magnetic_state` に `af` を指定しています。こうすると、スピン反転を伴う生成元を `af_generator` から読み込みます。この場合、スピン反転を伴う生成元は  $\{E|\frac{1}{2}\mathbf{t}_1 + \frac{1}{2}\mathbf{t}_2 + \frac{1}{2}\mathbf{t}_3\}$  です、それを `af_generator` に指定しています。

### 5.2.3 対称操作を自動的に決定する方法

`method` 変数に `automatic` を指定することで、対称性は自動的に決定されます。前述の生成元の指定は不要となりますが、タグ `tSPACE` 中の変数 `lattice_system` の指定は必須です。

```
symmetry{
  method = automatic
  tSPACE{
    lattice_system = facecentered
```

```

    !{rhombohedral|hexagonal|primitive|facecentered|bodycentered|basecentered}
  }
}

```

### 5.3 反転対称性がある場合

反転対称がある場合，これを考慮する事により，波動関数の計算に必要なメモリーサイズを低減させる事ができます。

#### 5.3.1 指定しないとき

変数 `sw_inversion` に `off` を入力します。このとき，Si 結晶の場合の `atom_list` は，以下のようになります。

```

atom_list{
  atoms{
    !#tag rx      ry      rz      element
          0.125   0.125   0.125    Si
          -0.125  -0.125  -0.125    Si
  }
}

```

#### 5.3.2 指定するとき

変数 `sw_inversion` に `on` を入力します。この場合，`atom_list` における原子座標の指定は，すでに第 3.1 節でも見てきたように，

```

atom_list{
  coordinate_system = internal ! {cartesian|internal}
  atoms{
    #units !{angstrom|bohr}
    #tag rx      ry      rz      weight  element  mobile
          0.125   0.125   0.125    2        Si        1
  }
}

```

として行ないます。反転対称により，自分自身のミラーとなる原子があるために，`weight` の値で示される各原子の重みづけは，2 となっています。

第 5.2 節で指定した対称群に反転対称操作が含まれる場合，この `option` を指定することを推奨します。なお，原子座標を入力する場合反転対称操作の中心は，原点であることにご注意ください。

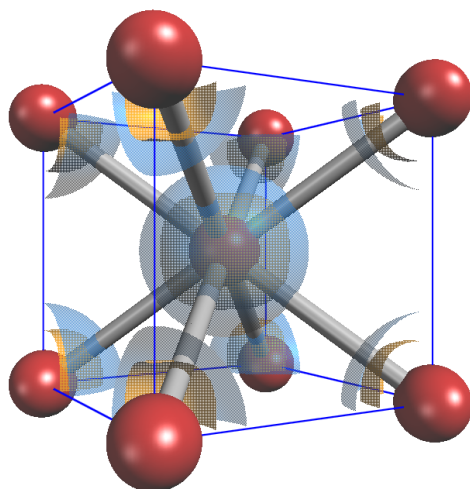


図 5: Fe の電荷密度分布図。青色とオレンジ色の面は、自発磁化により生じた、多数派スピンと少数派スピンによる電荷密度分布の等値面を表す。

## 6 計算例 3：スピン分極がある場合 (Fe)

強磁性体を扱う場合には、スピン分極を考慮しなければなりません。ここでは鉄を例に取り上げます (図 5 参照)。

`samples` のディレクトリー `bcc_Fe` の下の `scf` というディレクトリーに入ると、入力ファイル `input_scf_Fe.data` が見つかるはずです。その内容は以下の通りです。

```
Control{
    condition = initial
    cpumax = 3 hour
    max_iteration = 250
}

accuracy{
    cutoff_wf = 25 rydberg
    cutoff_cd = 225.00 rydberg
    num_bands = 20
    ksampling{
        method = mesh
        mesh{ nx = 10, ny = 10, nz = 10 }
    }
    smearing{
        method = tetrahedral
    }
    xctype = ggapbe
    scf_convergence{
        delta_total_energy = 1.e-10 hartree
    }

    initial_wavefunctions = matrix_diagon
    matrix_diagon{
        cutoff_wf = 12.25 rydberg
    }
}
```

```

}

structure{
    unit_cell_type = Bravais
    unit_cell{
        #units angstrom
        a = 2.845, b = 2.845, c = 2.845
        alpha = 90, beta = 90, gamma = 90
    }

    symmetry{
        crystal_structure = bcc
    }

    magnetic_state = ferro

    atom_list{
        atoms{
            !#tag rx ry rz element
            0.000 0.000 0.000 Fe
        }
    }
    element_list{ !#tag element atomicnumber zeta dev
                  Fe 26 0.275 1.5 }
}

wavefunction_solver{
    solvers{
        #tag sol till_n prec submat cmix
        davidson +2 off off 1
        rmm3 -1 on on 1
    }
    rmm{
        edelta_change_to_rmm = 5.0e-3
    }
    submat{
        before_renewal=on
    }
}

charge_mixing{
    sw_recomposing=on
    spin_density_mixfactor = 2
    mixing_methods{
        !#tag no method rmxs rmxe prec istr nbmix update
        1 broyden2 0.40 0.40 on 3 15 RENEW
    }
    spin_density{
        sw_force_simple_mixing = off
    }
}

Postprocessing{
    dos{
        sw_dos = ON
    }
}

```

```

        method = tetrahedral
        deltaE = 1.e-4 hartree
        nwd_dos_window_width = 10
    }
    charge{
        sw_charge_rspace      = OFF
        filetype              = cube
        title = "This is a title line for FM bcc Fe"
    }
}

printlevel{
    base = 1
}

```

## 6.1 スピン自由度の指定方法

強磁性体を扱う場合には、入力ファイルの `magnetic_state` の部分を、次のように記述します。

```

structure{
    magnetic_state = ferro    !{para|antiferro|ferro}
}

```

反強磁性体を扱う場合には、同様に、`magnetic_state = antiferro` とします。antiferro は af と略すことも可能です。さらに、各原子のスピン分極の初期値を指定する必要があります。入力ファイルにある、

```

        element_list{ #tag element  atomicnumber      zeta  dev
                        Fe            26              0.275  1.5
        }

```

の `zeta = 0.275` という変数の値が、アップ・スピンとダウン・スピンの密度の差を表す、スピン分極  $\zeta = (n_{\uparrow} - n_{\downarrow}) / (n_{\uparrow} + n_{\downarrow})$  の初期値を示しています。

## 6.2 スピン分極を固定する方法

通常スピン分極を変化させながら SCF 計算は進行しますが、以下の入力項目を付加することによってスピン分極を固定した計算を実行することも可能です。

```

structure{
    ...
    ...
    magnetic_state = ferro
    ferromagnetic_state{
        sw_fix_total_spin = on
        total_spin = 0.275
    }
}

```

スピン分極を固定した計算の方が、スピン分極を変化させる計算よりも高速に収束することが期待できます。

## 6.3 スピンを考慮している場合の SCF 計算の収束性

スピンを考慮している場合、SCF 計算は考慮していない場合と比較して収束しづらくなります。この例題では電荷密度混合法を次のように設定することによって比較的少ない計算回数（20 回程度）で収束へ至らせることができます。

```

charge_mixing{
  sw_recomposing=on
  spin_density_mixfactor = 2
  mixing_methods{
    !#tag no    method  rmxs  rmxe  prec istr  nbmix  update
        1  broyden2  0.40  0.40  on   3     15    RENEW
  }
}

```

sw\_recomposing を on とすることによって、電荷密度を和と差に分けて混合するようになります。また, spin\_density\_mixfactor によって和と差の混合比を変更しています。詳細は 8.3.3 を参照してください。

## 6.4 スピン分極の変化

スピン分極の時間変化は、ログファイルに書き込まれていますので、

```
% grep charge output000 | grep NEW | more
```

として確認することができます。その結果は、

```

!*--- input-file style = NEW
!NEW total charge (UP, DOWN, SUM) =      4.94504088 (+)      3.05495912 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      4.87168481 (+)      3.12831519 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      4.82057149 (+)      3.17942851 (=)      8.00000000
.....
.....
!NEW total charge (UP, DOWN, SUM) =      5.11279996 (+)      2.88720004 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      5.11279994 (+)      2.88720006 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      5.11279989 (+)      2.88720011 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      5.11279984 (+)      2.88720016 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      5.11279981 (+)      2.88720019 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      5.11279979 (+)      2.88720021 (=)      8.00000000

```

となります。ここで、スピン分極の定義  $\zeta = (n_{\uparrow} - n_{\downarrow}) / (n_{\uparrow} + n_{\downarrow})$  を使うと、これが  $\zeta = 0.2782$  という値に収束していることが分かります。

また、

```
% grep charge output000 | more
```

とすると、更新の前後での電荷分布の変化が確認できます。

```

F_CHGT      (= ./nfcharge.data      ) not existing
F_CHGT      = ./nfcharge.data      , newly opened
!** --- charge preconditioning ---
!** sw_charge_rspace      =      0
!** charge_filetype      =      1
!** charge_title      =
!** deviation( 1) of the Gauss. distrib. func. for the initial charge construction =      1.50000
F_CHGT      = ./nfcharge.data
F_CHGT      = ./nfcharge.data
!OLD total charge (UP, DOWN, SUM) =      5.10000000 (+)      2.90000000 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      4.94504088 (+)      3.05495912 (=)      8.00000000
!OLD total charge (UP, DOWN, SUM) =      4.94504088 (+)      3.05495912 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      4.87168481 (+)      3.12831519 (=)      8.00000000
!OLD total charge (UP, DOWN, SUM) =      4.87168481 (+)      3.12831519 (=)      8.00000000
!NEW total charge (UP, DOWN, SUM) =      4.82057149 (+)      3.17942851 (=)      8.00000000
.....
.....
.....

```



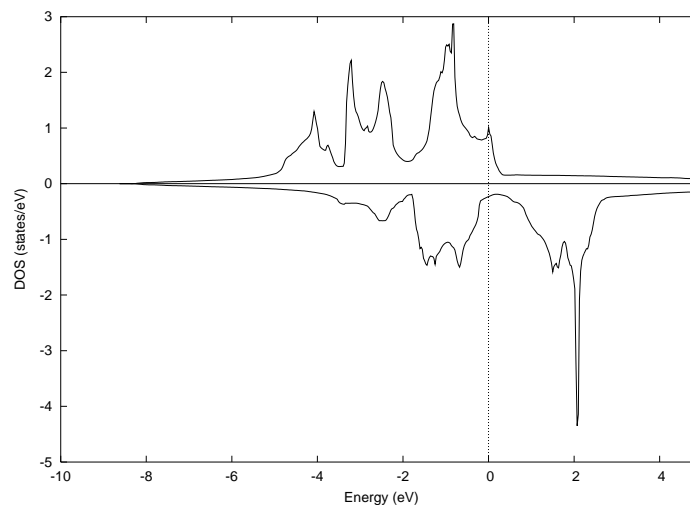


図 6: Fe の状態密度図。点線はフェルミ・レベルを表す。上側が多数派スピン状態，下側が少数派スピン状態の密度に対応

### 6.5 cube ファイル形式による電荷密度分布の出力

入力ファイルの最後にある，後処理 (postprocessing) というタグの中の charge の変数 `sw_charge_rspace` への入力値を ON にすると，スピンのアップとダウンに分極した，それぞれの電荷密度分布の cube ファイルが出力されます。その結果をもとに描いたのが，図 5 の電荷密度分布図です。

### 6.6 状態密度 と バンド構造

図 1 で示したように，tools に含まれている Perl スクリプトである `dos.pl` と `band.pl` を使うと，状態密度図とバンド構造を描くことが可能です。

まず，

```
% cd ../dos
```

として，`dos` という名前のディレクトリーに移ります。そこで `ekcal` による計算を行ない，その出力ファイルである `dos.data` に対して，tools にある `dos.pl` を使って，

```
% ../../tools/bin/dos.pl dos.data -erange=-E1,E2 -with_fermi
```

とすると，描画するエネルギー範囲の最小値が `E1`，最大値が `E2` で，フェルミ・レベルが点線で描かれた図が得られます (チュートリアル第 4.3 節を参照)。

ここでは，`E1 = -10` と `E2 = 5` を入力すると，図 6 にあるような状態密度図が得られます。

次に，

```
% cd ../band
```

として，`band` という名前のディレクトリーに移ります。ここで再び，`ekcal` による計算を行ない，それにより得られる出力ファイル `nfenergy.dat` に対して，tools にある Perl スクリプト `band.pl` を使って，前述と同じエネルギー幅で，

```
% ../../tools/bin/band.pl nfenergy.data ../tools/bandkpt_bcc_ghnpgn.in
-erange=-10,5 -with_fermi
```

というデータ処理を施すと，Postscript 形式のバンド構造図である `band.structure.eps` が得られます。この結果を，図 7 に示します。

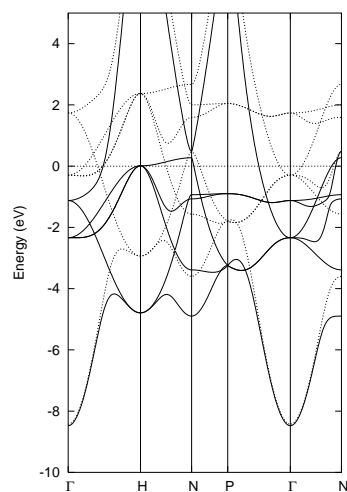


図 7: Fe のバンド構造。実線と点線が、それぞれ、多数派スピン状態と少数派スピン状態に対応。

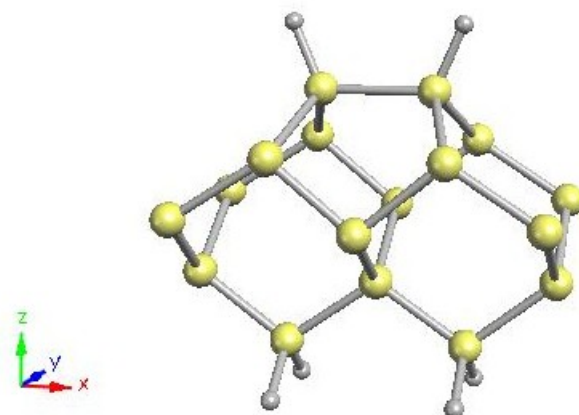


図 8: 水素終端した Si(001)- $p(2 \times 1)$  表面の構造図。スラブの下側の Si 原子のボンドは、仮想的な水素原子で終端されている。

## 7 計算例 4 : 水素終端 Si(001) 表面

複数の元素を同時に扱う場合の、入力ファイルの方法について、第 4.9 節で説明しました。ここでは、水素終端された Si(001) 表面を例に取り上げて、この方法について更に詳しく説明します。

この構造の計算には、図 8 に示されるような スラブ模型を使います。

この例で使用する file\_names.data を見ると、

```
&fnames
  F_INP      = './input_SiH2x1.data'
  F_POT(1)   = '../pp/Si_ldapw91_nc_01.pp'
  F_POT(2)   = '../pp/H_ldapw91_nc_01.pp'
  .....
&end
```

これから、F\_POT(1) と F\_POT(2) という 2 つの擬ポテンシャルが、Si 原子と H 原子のために使われていることが分かります。

```
accuracy{
  cutoff_wf = 15.00 rydberg
  cutoff_cd = 60.00 rydberg
  num_bands = 25
  ksampling{
    method = monk ! {mesh|file|directin|gamma}
    mesh{ nx = 2, ny = 4, nz = 1 }
    kshift{ k1 = 0.5, k2 = 0.5, k3 = 0.0 }
  }
  .....
}
```

この例では、構造模型として slab 模型を用いているため、 $k$  点のサンプリングは、 $k_z$  方向には 1 点だけを取っています。

```
structure{
  unit_cell_type = primitive
  unit_cell{
    a_vector = 14.512    0.000    0.000
    b_vector =  0.000    7.256    0.000
```

```

        c_vector =    0.000        0.000        30.784
    }
    symmetry{}

    magnetic_state = para    !{para|af|ferro}

    atom_list{
        coordinate_system = internal
        atoms{
            #default weight = 1, element = Si, mobile = 0
            #tag    rx        ry        rz        element
                0.26177    0.50000    0.65651        H
                0.73823    0.50000    0.65643        H
                0.34138    0.50000    0.56971
                0.65858    0.50000    0.56966
                0.26229    0.00000    0.49388
                0.73763    0.00000    0.49385
                0.00000    0.00000    0.41498
                0.50000    0.00000    0.40298
                0.00000    0.50000    0.32769
                0.50000    0.50000    0.32150
                0.25000    0.50000    0.24167
                0.75000    0.50000    0.24167
                0.25000    0.20000    0.18269        H
                0.25000    0.80000    0.18269        H
                0.75000    0.20000    0.18269        H
                0.75000    0.80000    0.18269        H
        }
    }
}

postprocessing{
    charge{
        sw_charge_rspace    = ON
        filetype            = cube    !{cube|density_only}
        title    = "Si(001) p(2x1) surface terminated by H atoms"
    }
}

```

タグ名 `atoms` の中で、デフォルト値として元素名を `Si` に設定しているので、変数 `element` に `H` と入力している以外の原子の元素名は `Si` になります。また、やはりデフォルト値として `mobile = 0` としているので、全ての原子の座標位置は固定されています。

```
% grep TOTAL output000
```

として、全エネルギーの収束状況を確認すると、以下のような結果が得られます。

```

TOTAL ENERGY FOR    1 -TH ITER=    -39.848530200757    edel =    -0.398485D+02
TOTAL ENERGY FOR    2 -TH ITER=    -41.609723584621    edel =    -0.176119D+01
TOTAL ENERGY FOR    3 -TH ITER=    -42.187351762815    edel =    -0.577628D+00
TOTAL ENERGY FOR    4 -TH ITER=    -42.567052165092    edel =    -0.379700D+00
TOTAL ENERGY FOR    5 -TH ITER=    -42.671401804133    edel =    -0.104350D+00
TOTAL ENERGY FOR    6 -TH ITER=    -42.830706589667    edel =    -0.159305D+00
TOTAL ENERGY FOR    7 -TH ITER=    -42.879741013145    edel =    -0.490344D-01
TOTAL ENERGY FOR    8 -TH ITER=    -42.901690423569    edel =    -0.219494D-01
TOTAL ENERGY FOR    9 -TH ITER=    -42.907076085380    edel =    -0.538566D-02
TOTAL ENERGY FOR   10 -TH ITER=    -42.925233423911    edel =    -0.181573D-01

```

```

.....
.....
TOTAL ENERGY FOR 65 -TH ITER= -42.965049366619 edel = -0.225628D-05
TOTAL ENERGY FOR 66 -TH ITER= -42.965049639587 edel = -0.272968D-06
TOTAL ENERGY FOR 67 -TH ITER= -42.965054943637 edel = -0.530405D-05
TOTAL ENERGY FOR 68 -TH ITER= -42.965059258453 edel = -0.431482D-05
TOTAL ENERGY FOR 69 -TH ITER= -42.965063421159 edel = -0.416271D-05
TOTAL ENERGY FOR 70 -TH ITER= -42.965066446217 edel = -0.302506D-05
TOTAL ENERGY FOR 71 -TH ITER= -42.965070113202 edel = -0.366699D-05
TOTAL ENERGY FOR 72 -TH ITER= -42.965071480045 edel = -0.136684D-05
TOTAL ENERGY FOR 73 -TH ITER= -42.965071551890 edel = -0.718448D-07
TOTAL ENERGY FOR 74 -TH ITER= -42.965071603914 edel = -0.520233D-07
TOTAL ENERGY FOR 75 -TH ITER= -42.965071513030 edel = 0.908839D-07

```

この例題は、固体表面構造に対するエネルギー計算だけを目的にしていますが、もし原子位置の緩和過程の計算を行なうなら、

```

atoms{
  #default weight = 1, element = Si, mobile = 1
  #tag   rx      ry      rz      element  mobile
        0.26177  0.50000  0.65651    H
        0.73823  0.50000  0.65643    H
        0.34138  0.50000  0.56971
        0.65858  0.50000  0.56966
        0.26229  0.00000  0.49388
        0.73763  0.00000  0.49385
        0.00000  0.00000  0.41498
        0.50000  0.00000  0.40298
        0.00000  0.50000  0.32769
        0.50000  0.50000  0.32150
        0.25000  0.50000  0.24167    *      0
        0.75000  0.50000  0.24167    *      0
        0.25000  0.20000  0.18269    H      0
        0.25000  0.80000  0.18269    H      0
        0.75000  0.20000  0.18269    H      0
        0.75000  0.80000  0.18269    H      0
}

```

として、下端の仮想水素とそれらと結合した Si 原子を固定し、それら以外の原子を可動 (mobile = 1) に変えてやる必要があります。

Si(001) 表面のバククルしたダイマーの安定構造は  $p(2 \times 1)$  ではなく  $c(4 \times 2)$  ですが、この構造を再現するには、Si ダイマーをもう一つ増やすなどして、最上層に位置する Si ダイマーの総数を偶数個にしなければなりません。

尚、構造緩和の計算方法については、本マニュアルの第 8 節をご参照ください。

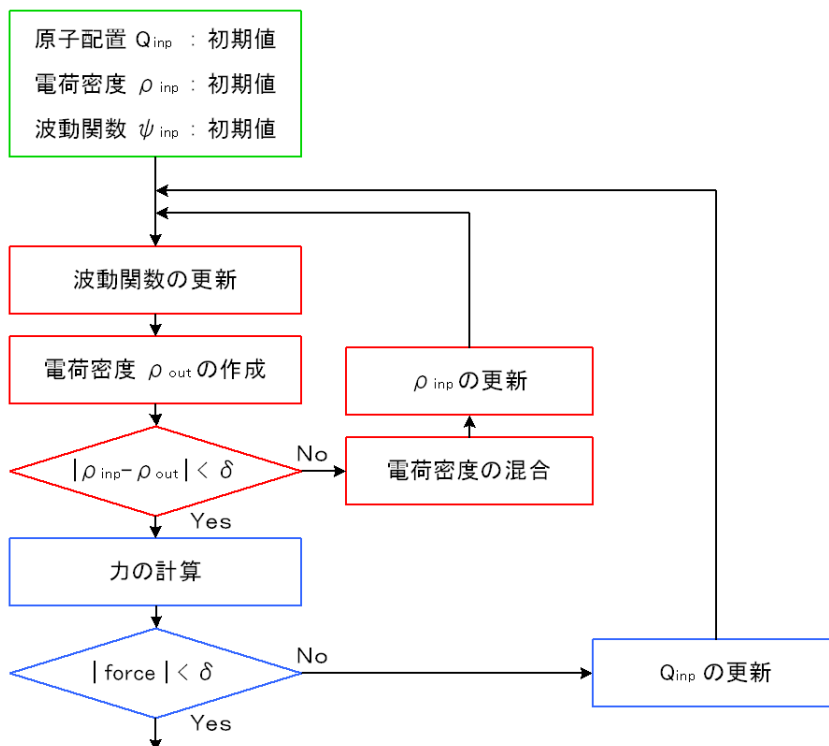


図 9: 最適化計算のフローチャート

## 8 PHASE における最適化計算

### 8.1 作業フロー

チュートリアル の例題 3 で、原子構造の最適化を行いました。この時の PHASE 内の作業を図 9 に示します。図中の波動関数の更新の過程で、Kohn-Sham 方程式

$$(H_{KS}(\rho_{inp}) - \epsilon_i)\psi_i = 0, \quad (3)$$

を解いています。

ある試行の波動関数が与えられ、

$$\Delta_i = (H_{KS}(\rho_{inp}) - \epsilon_i)\psi_i \quad (4)$$

の演算を繰り返し行うことにより、(3) 式の解が得られることになります。その際、 $\Delta_i$  はエネルギー  $\epsilon_i$  の波動関数  $\psi_i$  に対する勾配と解釈する事ができるので、この勾配が 0 に近づくように波動関数が更新されます。フローチャート図 9 中の電荷密度の作成の過程では、更新した波動関数から、新しい電荷密度  $\rho$  が、以下の処方で与えられます。

$$\rho_{out} = 2 \sum_{occ.} |\psi_i|^2, \quad (5)$$

フローチャート図中の内側のループでは、入力  $\rho_{inp}$  と新しい  $\rho_{out}$  が一致するまで計算が行われます。この作業は SCF(自己無動着場) 計算と呼ばれています。外側のループでは、与えられた原子配置に対して力の計算が行われ、この力が 0 になるような原子配置に到達するまで、計算が続行されます。

### 8.2 電子状態最適化ループ

SCF ループにおける電荷密度の混合においては、単純混合法や Broyden 法が選択できます。波動関数の更新では、多様なオプションが用意されています。表 2 中の wavefunction\_solver の項目を参照して下さい。ここで、

LM+MSD 法は、(4) 式の  $\Delta_i$  の値が大きくな時に有効な方法です。一方、 $\Delta_i$  の値が小さな時に RMM 法を用いると、収束が速くなります。ただし、 $\Delta$  の大きな場合、RMM 法は不安定になる場合もあります。また、PHASE では、繰り返しの方法ではなく、行列対角化により、(3) 式を解くためのオプション (Matrix diagonalization) も用意されています。

## 8.3 波動関数 solver と電荷密度混合のオプション選択の仕方

### 8.3.1 波動関数 solver

波動関数 solver は、`wavefunction_solver` タグで設定します。`wavefunction_solver` タグの例を以下に示します。なお、波動関数の初期状態は、行列対角化、乱数、原子軌道から選択することができますが、それらの値は、`accuracy` タグ中の `initial_wavefunctions` タグで設定します。

```
wavefunction_solver{
  solvers{
    #tag      sol      till_n  dts  dte  itr  var      prec  cmix
          lmMSD      5      0.2  1.0  40   linear  on     1
          RMM3      -1      0.2  0.6  20   linear  on     2
  }
  line_minimization{
    dt_lower_critical  = 0.5
    dt_upper_critical  = 5.0
    delta_lmdenom      = 1e-12
  }
  rmm{
    imGSrmm            = 1
    rr_Critical_Value   = 1.e-15
    edelta_change_to_rmm = 1.0e-4 hartree
  }
}
```

`solvers` タグでは、波動関数の更新方法を設定します。`id` は、ソルバーが選択される順番を表し、`sol` は、波動関数の更新方法を指定します。その繰り返し数が `till_n` に至るまで、`sol` で指定された波動関数の更新方法を用います。次に、`dts`, `dte`, `itr`, `var` は、波動関数更新の時間刻み幅を指定します。刻み幅は、開始時に `dts` で、`itr` 回の繰り返し後に `dte` となるように `var` 形式で補完されます。`itr` 回の繰り返し以降は、一定値 `dte` となります。次に、`prec` は、波動関数の更新で前処理を行うかどうかのフラグです。最後に `cmix` は、電荷密度の混合方法を指定する番号です。`charge_mixing` タグの `mixing_method` タグの中の変数名 `no` で指定される電荷密度の混合方法が用いられます。

`line_minimization` タグでは、lmMSD 法 (MSD 法+一次元探索) において、波動関数の最適更新時間を探索するのに関係した時間 (`dt_lower_critical` と `dt_upper_critical`) を設定できます。一次元探索法では、iteration ごとに時間と全エネルギーの 3 つの組である  $(t_0, e_0)$ ,  $(t_0+dt, e_1)$ ,  $(t_0+2dt, e_2)$  から最適更新時間  $dt_{opt}$  を二次関数フィッティングにより決定します。ここで  $e_0$  は、前の iteration で得られた全エネルギーを指します。この  $dt_{opt}$  を使って波動関数を更新し、電荷密度分布、全エネルギーを計算します。

次の iteration の `dt` には、原則として  $dt_{opt}$  を使いますが、 $dt_{opt}$  が大きすぎたり小さすぎたりすると、二次関数を使った最適更新時間  $dt_{opt}$  の探索ができなくなることがあります。そこで、PHASE の中では `dt` の値の下限 (`dt_lower_critical`) と上限 (`dt_upper_critical`) を設定してあります。これらの値の既定値は、それぞれ、0.05 (au.time) と 2.0 となっています。

`rmm` タグでは、グラム・シュミットの直交化法、バンド毎の収束条件、RMM の開始条件を設定します。`imGSrmm` は、グラム・シュミットの直交化法を行う周期を指定します。この値が 1 の場合には、更新のたび毎に直交化を行います。`rr_Critical_Value` は、バンド毎の収束条件を指定します。各バンドの残差がこの設定値より小さい場合には、そのバンドの更新は行いません。`edelta_change_to_rmm` は、波動関数の更新方法を RMM に変えるときの条件を指定します。全エネルギーの変化がこの値より小さい場合に、波動関数の更新方法を RMM に切り替えます。

上記の例では、5 回目まで、lmMSD 法を用いて波動関数の更新を行い、6 回目以降で、全エネルギーの変化が  $1.0e-4$  hartree 以下になった時に、波動関数の更新方法が RMM3 法に切り替わります。1 行目の lmMSD 法の設定について詳しく説明すると、波動関数の更新の時間刻み幅は、開始時に  $dts = 0.2$  であり、繰り返し毎に  $(dte - dts)/itr = (1.0 - 0.2)/40 = 0.02$  ずつ増加していき、繰り返し数が 41 以降では、 $dte = 1.0$  となります。前処理用の

変数 `prec` の値が `on` になっているので、波動関数の更新時に前処理を行います。電荷密度混合は、`charge_mixing` タグの第 2 タグである `mixing_methods` の、第 1 番目の設定が用いられます。

Davidson 法を使用するときには注意が必要です。現在実装されている Davidson 法のアルゴリズムでは、`prec` を `off` に設定した場合に正常に収束します。

### 8.3.2 電荷密度混合

電荷密度混合は、`charge_mixing` というタグで設定します。`charge_mixing` タグの例を以下に示します。

```
charge_mixing{
  mixing_methods{
    #tag    no  method    rmxs  rmxe  itr  var    prec  istr  nbmix  update
          1  broyden2  0.6   0.8   20   linear on    0     5     renew
          2  simple    0.2   0.5   20   linear on
  }
  charge_preconditioning{
    amix    = 0.90
    bmix    = -1
  }
}
```

`mixing_methods` タグでは、電荷密度の混合方法を設定します。変数 `no` は、波動関数の更新の設定中における `wavefunction_solver` というタグの第 2 タグである `solvers` の中の、変数 `cmix` で指定されている番号に対応しています。第 2 変数である `method` は、電荷密度の混合方法を指定します。その次の 4 つの変数 `rmxs`, `rmxe`, `itr`, `var` は、電荷密度の混合比を指定します。混合比は、開始時に `rmxs` で、`itr` 回の繰り返し後に `rmxe` となるように `var` 形式で補完されます。`itr` 回の繰り返し以降は、一定値 `rmxe` に留まります。変数 `prec` は、波動関数の更新で前処理を行うかどうかを指定するフラグです。残りの 3 つの変数 `istr`, `nbmix`, `update` は、Broyden 法で使用する変数です。`istr` は単純混合法から broyden2 法に切り替えるまでの繰り返し回数を指定します。`nbmix` は、過去の電荷密度の保持数を指定します。変数 `update` を使うと、過去の電荷密度の保存方法を指定し、`anew` と `renew` を設定することができます。`anew` の場合は、`nbmix` 個の電荷密度データになると、全ての電荷密度データを消してから、新しい電荷密度を保存していきます。`renew` の場合は、`nbmix` 個の中の最も古い電荷密度データを、新しい電荷密度データに書き換えます。

`charge_preconditioning` タグでは、前処理の係数を設定します。前処理が有効の場合には、次の式を使って、 $G$  の成分毎の混合比を変えます。

$$\rho_{new}(G) \leftarrow (1 - f(G))\rho_{old}(G) + f(G)\rho_{new}(G), \quad (6)$$

$$f(G) = \frac{rmx * amix}{1 + (G_0^2/G^2)} \quad (7)$$

$$G_0 = bmix * G_{min} \quad (8)$$

ここで  $G_{min}$  は、原点以外の  $G$  の最小値を指します。また、 $bmix < 0$  のときは、 $G_0 = 1.5$  とします。 $bmix$  は、詳細な設定が必要な場合に用いますが、通常は  $-1$  を入力して、デフォルト値を用いるように指定しておきます。

上記の例では、1 番目の電荷密度の混合方法として Broyden2 法が、2 番目の電荷密度の混合方法として単純混合法が使用されます。1 番目の Broyden2 法の設定について詳しく説明すると、電荷密度の混合比は、開始時に  $rmxs = 0.6$  であり、繰り返し毎に  $(rmxe - rmxs)/itr = (0.8 - 0.6)/20 = 0.01$  ずつ増加していき、繰り返し数が 21 回以降では、 $rmxe = 0.8$  となります。さらに、変数 `prec` の値が `on` になっているので、波動関数の更新時に、前処理を行います。`istr = 0` であることから、計算開始時から Broyden2 法が用いられます。`nbmix = 5` 回の過去の電荷密度データを保存して、電荷密度の保存方法は、`renew` を用います。

### 8.3.3 収束を加速させるテクニック

ここでは、なかなか収束しない場合について試すことのできるテクニックを紹介します。



**部分空間対角化** 部分空間対角化は、デフォルトでは無効の状態です。有効にすると1回の計算にかかる時間は増えますが、多くの場合収束が加速されます。部分空間対角化を有効にするには、変数 `submat` を定義してその値を `on` とします。

```
wavefunction_solver{
  solvers{
    #tag      sol      till_n  dts  dte  itr  var      prec  cmix  submat
          lmMSD      -1      0.2  1.0  40   linear  on    1    on
  }
}
```

部分空間対角化の適用を、波動関数を更新する前に適用するか後に適用するかによって収束の振る舞い変化します。これは、特に RMM 法を利用している場合に大きな影響を与えます。デフォルトの振る舞いでは波動関数更新後に部分空間対角化が適用されますが、波動関数更新前にする場合には以下のように変数 `before_renewal` を `on` とします。

```
wavefunction_solver{
  solvers{
    #tag      sol      till_n  dts  dte  itr  var      prec  cmix  submat
          lmMSD      -1      0.2  1.0  40   linear  on    1    on
  }
  submat{
    before_renewal=on
  }
}
```

また、部分空間対角化はバンド数が多い方がより有効に作用します。バンド数を増やせばそれだけ計算量も増えますが、この効果によって全体の計算時間は短くなる場合もあります。

**SCF 計算をある回数で打ち切る方法** 初期の原子配置が安定な原子配置から遠い場合、SCF 計算を収束させるのに多くの繰り返し計算が必要となる場合があります。このような場合は、たとえ電子状態が十分に収束していなくとも構造最適化をすすめることによって結果的に正しい解へより少ない計算時間で到達することができる場合があります。そこで、入力指定の収束条件を満たしていなくとも収束したとみなし、構造最適化を進める機能が PHASE には備わっています。この機能を利用するためには、`control` ブロックの下で `max_scf_iteration` 変数を設定します。

```
control{
  ...
  max_scf_iteration = 50
}
```

この例では、50 回の SCF 計算を行っても収束判定を満たせなかった場合、その時点で至っている電子状態を利用して原子間力を計算し、構造最適化を進行させます。

**電荷密度の差の混合比を変更する方法** スピンを考慮している場合、電荷密度混合は全電荷とスピン電荷密度（アップスピンの電荷密度とダウンスピンの電荷密度の差）に分離して混合します。全電荷とスピン電荷の混合比をそれぞれ違う値に設定することが可能です。このような設定を行うには、下記の要領で `spin_density_mixfactor` 変数を定義します。

```
charge_mixing{
  spin_density_mixfactor = 4
  mixing_methods{
    #tag      no  method      rmxs  rmxe  prec  istr  nbmix  update
          1   broyden2  0.1  0.1  on   3  15  renew
  }
}
```

この例の場合, `spin_density_mixfactor` は 4 であり, 電荷密度の差の混合比は  $0.1 \times 4 = 0.4$  という値が採用されます。

全電荷とスピン電荷を混合するのではなくアップスピンの電荷密度とダウンスピンの電荷密度を直接混合する場合, 以下の要領で `sw_recomposing` 変数に `off` を設定します。

```
charge_mixing{
    sw_recomposing = off
    ...
}
```

スピン電荷密度の混合に利用するアルゴリズムを変更する スピン電荷密度に対して, 強制的に単純混合法を採用することも可能です。このような設定は, 以下のように `spin_density` ブロックを作成し, `sw_force_simple_mixing` 変数を定義しその値を `on` とします。

```
charge_mixing{
    sw_recomposing=on
    spin_density_mixfactor = 4
    mixing_methods{
        #tag    no method    rmxs  rmxe  prec  istr  nbmix  update
            1    broyden2  0.1 0.1 on 3 15 renew
    }
    spin_density{
        sw_force_simple_mixing = on
    }
}
```

スピンを固定する方法 一定の間スピンを固定して SCF 計算を行うと収束性が改善する場合があります。この設定は, 下記の要領で `structure` ブロックの下に `ferromagnetic_state` ブロックを作成し行います。

```
structure{
    ...
    ferromagnetic_state{
        sw_fix_total_spin = on
        spin_fix_period = INITIALLY
        total_spin = 1.0
    }
    ...
}
```

`ferromagnetic_state` ブロックでは以下の変数を利用することができます。

- `sw_fix_total_spin` : “on” とした場合, スピンを固定した計算を行います。
- `spin_fix_period` : スピン固定の方法を指定します。“INITIALLY” と指定した場合, SCF 計算の初期は固定し, すこしずつ拘束を外していきます。“WHOLE” と指定した場合, 計算終了までスピンを固定します。整数を指定した場合, その回数だけ固定しあとは通常の計算を行います。
- `total_spin` : アップスピンとダウンスピンの差を指定します。単位胞全体の値を指定してください。

欠損電荷を混合する方法 PAW 法を利用している場合, 欠損電荷の混合が行われます。DFT+U 法を利用している場合, 占有行列の混合が行われますが, これも実質上は欠損電荷の混合をおこなっていることと同等です。この混合に対して通常の電荷密度と同様のアルゴリズムで混合させるには, 以下のように `charge_mixing` ブロックに `sw_mix_charge_hardpart` 変数を定義し, その値を `on` にします。

```
charge_density{
    ...
    sw_mix_charge_hardpart = on
    ...
}
```

このように設定することによって, PAW 法および DFT+U 法利用時の収束性が向上する場合があります。

## 8.4 SCF 計算ベンチマークテスト

### 8.4.1 FCC-Cu

FCC-Cu について、各波動関数ソルバーがどの程度のパフォーマンスを示すか確かめてみましょう。なお、ここで紹介する計算の入力データは、以下のディレクトリーのサブディレクトリーにあります。

`samples/sol_cmix_test/Cu`

まず、波動関数ソルバー以外の入力データです。

```
Control{
  condition = initial
  cpumax = 1 day
}
accuracy{
  cutoff_wf = 25.0 rydberg
  cutoff_cd = 225.0 rydberg
  num_bands = 10
  ksampling{
    mesh{
      nx = 10
      ny = 10
      nz = 10
    }
  }
  scf_convergence{
    delta_total_energy = 1.e-10 hartree
    succession = 3
  }
  initial_wavefunctions = atomic_orbitals
  initial_charge_density = atomic_charge_density
}
structure{
  unit_cell_type = primitive
  unit_cell{
    !#units bohr
    a_vector = 0.0000000 3.4704637 3.4704637
    b_vector = 3.4704637 0.0000000 3.4704637
    c_vector = 3.4704637 3.4704637 0.0000000
  }
  symmetry{
    method = automatic
    tspace{
      lattice_system = fcc
    }
    sw_inversion = on
  }
  atom_list{
    atoms{
      !#tag rx ry rz weight element mobile
      0.000 0.000 0.000 1 Cu 0
    }
  }
  element_list{
    #tag element atomicnumber
    Cu 29
  }
}
```

```

}
wavefunction_solver{
    次節参照
}
charge_mixing{
    mixing_methods{
        !#tag  method  rmxs  rmxe  itr  var  prec istr  nbmix  update
            broyden2  0.60  0.60  *    *    on   3    15    RENEW
    }
}
printlevel{
    base = 1
}

```

以下に波動関数ソルバーの設定例を示します。

#### 行列対角化

```

wavefunction_solver{
    solvers{
        !#tag  id sol  till_n  dts  dte  itr  var  prec cmix submat
            1  matrixdiagon  -1    *    *    *    *    *    on 1 off
    }
}

```

#### lm+msd, 部分空間対角化は波動関数更新後

```

wavefunction_solver{
    solvers{
        !#tag  id sol  till_n  dts  dte  itr  var  prec cmix submat
            1  lm+msd  1    *    *    *    *    *    on 1 on
    }
    submat{
        before_renewal=off
    }
}

```

#### lm+msd, 部分空間対角化は波動関数更新前

```

wavefunction_solver{
    solvers{
        !#tag  id sol  till_n  dts  dte  itr  var  prec cmix submat
            1  lm+msd  1    *    *    *    *    *    on 1 on
    }
    submat{
        before_renewal=on
    }
}

```

#### lm+msd → rmm3, 部分空間対角化は波動関数更新後

```

wavefunction_solver{
    solvers{
        !#tag  id sol  till_n  dts  dte  itr  var  prec cmix submat
            1  lm+msd  1    *    *    *    *    *    on 1 on
    }
}

```

```

                2 rmm3      -1    *    *    *    *    on    1    on
    }
    rmm{
        edelta_change_to_rmm = 5.0e-3
    }
    submat{
        before_renewal=off
    }
}

```

lm+msd → rmm3, 部分空間対角化は波動関数更新前

```

wavefunction_solver{
    solvers{
        !#tag    id sol    till_n    dts    dte    itr    var    prec    cmix    submat
            1 lm+msd    1    *    *    *    *    on    1    on
            2 rmm3      -1    *    *    *    *    *    on    1    on
    }
    rmm{
        edelta_change_to_rmm = 5.0e-3
    }
    submat{
        before_renewal=on
    }
}

```

Davidson → rmm3, 部分空間対角化は波動関数更新後

```

wavefunction_solver{
    solvers{
        !#tag    id sol    till_n    dts    dte    itr    var    prec    cmix    submat
            1 davidson    1    *    *    *    *    off    1    off
            2 rmm3      -1    *    *    *    *    *    on    1    on
    }
    rmm{
        edelta_change_to_rmm = 5.0e-3
    }
    submat{
        before_renewal=off
    }
}

```

Davidson → rmm3, 部分空間対角化は波動関数更新前

```

wavefunction_solver{
    solvers{
        !#tag    id sol    till_n    dts    dte    itr    var    prec    cmix    submat
            1 davidson    1    *    *    *    *    off    1    off
            2 rmm3      -1    *    *    *    *    *    on    1    on
    }
    rmm{
        edelta_change_to_rmm = 5.0e-3
    }
    submat{
        before_renewal=on
    }
}

```

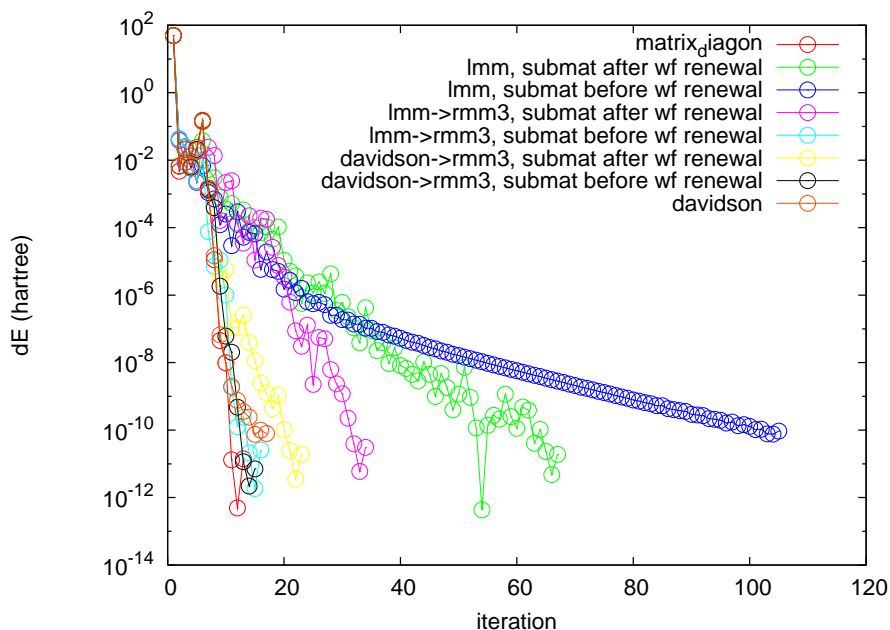


図 10: 波動関数ソルバーによる収束の速さの比較。

表 5: 波動関数ソルバーによる計算時間の比較。繰り返し回数は、電荷の混合の回数を指す。また、計算時間は Opteron 280 2.4GHz のクラスターマシンで、k 点 4 並列で行った場合の結果です。あくまで参照値とお考えください。

手法	繰り返し回数	計算時間
行列対角化法	13 回	19.2 秒
lm+msd, 部分空間対角化は波動関数更新後	67 回	22.2 秒
lm+msd, 部分空間対角化は波動関数更新前	105 回	32.4 秒
lm+msd → rmm3, 部分空間対角化は波動関数更新後	34 回	12.4 秒
lm+msd → rmm3, 部分空間対角化は波動関数更新前	16 回	8.4 秒
Davidson → rmm3, 部分空間対角化は波動関数更新後	23 回	11.2 秒
Davidson → rmm3, 部分空間対角化は波動関数更新前	15 回	9.5 秒
Davidson	17 回	11.8 秒

### Davidson

```

wavefunction_solver{
  solvers{
    !#tag  id sol  till_n  dts  dte  itr  var  prec cmix submat
    1  davidson  -1    *    *    *    *    off 1 off
  }
}

```

ベンチマークテストの計算結果を図 10 に、計算時間を表 5 に示します。Intel Fortran Compiler 11.1 for Linux でコンパイルして、2.4GHz の Opteron280 プロセッサを搭載したクラスターマシンにおいて k 点 4 並列で計算を行いました。

横軸は繰り返し回数で、縦軸は収束値からの相対的なエネルギーです。SCF 計算では、変分原理が成り立つので、エネルギーが低いほど、正確な値となります。行列対角化法は、繰り返し回数で見た収束は速いのですが、1 回あたりの計算量は一般に大きく、とくに、系のサイズが大きくなると、この手法は実際上適用できなくなります。全体的には、途中で rmm3 法へ移行するケースが速く収束しています。また、rmm3 法は部分空間対角化を波動関数更新前に適用する方がより速く収束しています。

計算する系に依存して、収束の速さ、安定性は変化します。そこで、その都度、最適な計算手法を選択する事をお勧めいたします。比較的安定で収束の速い手法として、LM+MSD RMM3 法, Davidson 法, Davidson

RMM3 法が多くの場合推奨されます。また, RMM3 法を利用する場合部分空間対角化は波動関数更新前に適用した方がよい場合が多いです。Davidson 法を利用する場合, 前処理 (precon) は off としておいた方がよいでしょう。

#### 8.4.2 Fe(100) 表面

次に, スピンを考慮した計算の例として Fe(100) 表面の計算例を紹介します。この問題では, 波動関数ソルバーは固定とし, 電荷密度混合法を変更したテストを行います。ここで紹介する計算の入力データは, 以下のディレクトリーのサブディレクトリーにあります。

`samples/sol_cmix_test/Fe100`

まず, 電荷密度混合法以外の入力データです。

```
control{
  condition = initial
  max_iteration = 200
}
accuracy{
  num_bands = 52
  ksampling{
    method=monk
    mesh{
      nx = 6
      ny = 6
      nz = 1
    }
  }
}
cutoff_wf = 30 rydberg
cutoff_cd = 300 rydberg
initial_wavefunctions = atomic_orbitals
initial_charge_density = atomic_charge_density
scf_convergence{
  delta_total_energy = 1e-9
  succession = 3
}
force_convergence{
  max_force = 0.0005 hartree/bohr
}
}
structure{
  atom_list{
    atoms{
      #tag    element    rx    ry    rz    mobile    weight
      Fe      0.5      0.5    0    off     1
      Fe      0        0    0.0948333333333333    off     2
      Fe      0        0    0.2845    off     2
      Fe      0.5      0.5    0.18966666666667    off     2
    }
  }
}
ferromagnetic_state{
  sw_fix_total_spin=off
  total_spin=14
  spin_fix_period=5
}
unit_cell{
  a_vector = 5.3762704477 0.0 0.0
  b_vector = 0.0 5.3762704477 0.0
}
```

```

    c_vector = 0.0 0.0 28.3458898822
}
element_list{
  #tag      element      atomicnumber      mass      zeta      deviation
        Fe        26        101802.230406        0.375        1.83
}
symmetry{
  method = automatic
  sw_inversion = on
}
magnetic_state=ferro
}
structure_evolution{
  method = gdiis
  gdiis{
    initial_method = cg
    c_forc2gdiis = 0.005 hartree/bohr
  }
}
wavefunction_solver{
  solvers{
    #tag      sol      till_n      prec      cmix      submat
        davidson      1      off      1      off
        rmm3          -1      on      1      on
  }
  rmm{
    edelta_change_to_rmm = 5e-3 hartree
  }
  submat{
    before_renewal = on
  }
}
charge_mixing{
  次節参照
}
printoutlevel{
  base = 1
}

```

以下に電荷密度混合法の設定例を示します。

電荷密度の和と差で異なる混合比を採用，混合アルゴリズムは Broyden2 法 (case0)

```

charge_mixing{
  spin_density_mixfactor=4
  mixing_methods{
    #tag      no      method      rmxs      rmxe      itr      var      prec      istr      nbmix      update
        1      broyden2      0.1      0.1      40      linear      on      3      5      renew
  }
}

```

電荷密度の和と差で異なる混合比を採用，混合アルゴリズムは Pulay 法 (case1)

```

charge_mixing{
  spin_density_mixfactor=4
  mixing_methods{
    #tag      no      method      rmxs      rmxe      itr      var      prec      istr      nbmix      update

```



表 6: Fe (100) 表面の収束に至るまでの SCF 計算回数と到達した全エネルギー

	SCF 回数	全エネルギー (ha.)
case0	36	-153.877775988322
case1	32	-153.877775991437
case2	34	-153.877775825592
case3	29	-153.877775990755

```

1    pulay    0.1    0.1    40    linear    on    3    15    renew
}
}

```

電荷密度の和と差で同じ混合比を採用，混合アルゴリズムは Broyden2 法 (case2)

```

charge_mixing{
  spin_density_mixfactor=1
  mixing_methods{
    #tag    no    method    rmxs    rmxe    itr    var    prec    istr    nbmix    update
    1      broyden2    0.1    0.1    40    linear    on    3    15    renew
  }
}

```

電荷密度の和と差で同じ混合比を採用，混合アルゴリズムは Pulay 法 (case3)

```

charge_mixing{
  spin_density_mixfactor=1
  mixing_methods{
    #tag    no    method    rmxs    rmxe    itr    var    prec    istr    nbmix    update
    1      pulay    0.1    0.1    40    linear    on    3    15    renew
  }
}

```

本ベンチマークの結果を，表 6 にまとめました。同じ電子状態へ収束していることを確認するため，得られた全エネルギーもあわせて表示しています。この例では，「電荷密度の和と差で同じ混合比を採用，混合法は Pulay 法」(case3) で最少の回数で収束解を得ることができました。多くの場合 case3 の設定によって少ない計算回数で収束解が得られますが，問題によっては Broyden2 法を採用した方が良い場合もありますし，電荷密度の差の混合比を大きくした方がよい場合もあります。さらに，スピンを考慮した計算の場合，初期スピン分極やスピンを固定するか否かなどによっても収束性は変化します。収束しづらい(しない)場合，上記の設定例などを参考に，最適な電荷密度混合法の設定を行ってください。

## 8.5 構造最適化ループ

第 9 図中の外側に位置するループを実行する際には，Quenched MD 法と，GDIIS 法，BFGS 法，CG 法が選択できます。これは，第 3 節で説明した入力ファイルの，structure\_evolution のタグを使って選択できます。

Quenched MD 法は，原子に働く力が比較的大きな時に用いられます。これに対して GDIIS 法や BFGS 法を使うと，力の大きさがある程度小さくなったときに，効率良く計算を収束させられます。CG 法を使うと確実に安定構造に到達できます。さらに，力の大きな時に quenched MD 法あるいは CG 法を用い，小さくなると GDIIS 法あるいは BFGS 法に切り替える計算を選択することもできます。計算の状況に応じ，これらの 4 つの方法から適宜選択し，原子構造の最適化を行なってください。

それぞれのアルゴリズムの性能を調べるため，以下の系でベンチマークテストを行いました。

- ケース 1 : cis 型のジクロロシクロヘキサン
- ケース 2 : ルチル型  $\text{TiO}_2$
- ケース 3 :  $\text{SiO}_2$

- ケース 4: Si(001) 表面

これらの入力は、以下のディレクトリー下のサブディレクトリーにあります。

`samples/strevl_test`

力の収束判定は、すべてのケースで  $10^{-4}$  hartree/bohr としました（この条件は、比較的厳しい収束判定条件です）。また、電子状態計算の収束判定はすべてのケースで  $10^{-10}$  hartree 1 回としました。原子配置の更新は、最大 200 回行い、それでも収束しないケースは未収束と見なしました。

各々の構造最適化のアルゴリズムは、以下のように設定しました。

### 8.5.1 quenched MD 法

```
structure_evolution{
  method = quench
}
```

### 8.5.2 cg 法

```
structure_evolution{
  method = cg
}
```

### 8.5.3 gdiis 法

```
structure_evolution{
  method = gdiis
  gdiis{
    initial_method = cg
    c_forc2gdiis = 0.01 hartree/bohr
  }
}
```

このように設定すると、まずは cg 法のアルゴリズムに従って最適化が進行します。原子に働く力の最大値が `c_forc2gdiis` で指定する値よりも小さくなった時点で gdiis 法に切り替わります。ただし、最初の 3 回は原子に働く力の最大値に関わらずに cg 法が採用されます。

### 8.5.4 bfgs 法

```
structure_evolution{
  method = bfgs
  gdiis{
    initial_method = cg
    c_forc2gdiis = 0.01 hartree/bohr
  }
}
```

このように設定すると、まずは cg 法のアルゴリズムに従って最適化が進行します。原子に働く力の最大値が `c_forc2gdiis` で指定する値よりも小さくなった時点で bfgs 法に切り替わります。ただし、最初の 3 回は原子に働く力の最大値に関わらずに cg 法が採用されます。bfgs 法の詳細設定は gdiis ブロックで行い、また各変数の意味も gdiis 法の場合と全く同じです。

ベンチマークの結果を表 7 にまとめました。この結果、quenched MD 法は収束が遅いことが分かります。今回のケースでは時間刻みとしてデフォルト値（100 au）を採用しましたが、この値を調整することによって収束が改善される可能性はあります。GDIIS 法は SiO<sub>2</sub> のケースでは最も速く収束していますが、それ以外のケースではあまり有効に作用していません。gdiis 法の振る舞いは、`c_forc2gdiis` パラメーターをより小さなものにしたり、電子状態計算の収束判定をより厳しくすることによって改善される可能性はあります。cg 法は比較的安定に最適化ができています。bfgs 法は、すべてのケースで収束し、また平均的に少ない回数で最適化を行うことができました。

表 7: 構造緩和法の違いによる収束にいたる回数の比較。200 回の更新を経ても力の最大値が  $10^{-4}$  以下とならなかったケースは未収束とした。ケース 1 がジクロロシクロヘキサン, ケース 2 がルチル型  $\text{TiO}_2$ , ケース 3 が  $\text{SiO}_2$ , ケース 4 が  $\text{Si}(001)$  表面に相当する。

	ケース 1	ケース 2	ケース 3	ケース 4
quenched MD	未収束	115	166	未収束
cg	195	62	28	124
GDIIS	未収束	71	13	176
BFGS	87	38	16	67

## 9 分子動力学シミュレーション

### 9.1 はじめに

PHASE は、構造最適化だけでなく原子に働く力を利用して分子動力学シミュレーションを行うことも可能です。本節はこの機能について説明します。なお、現バージョンの PHASE がサポートする統計集合は小正準集合 (エネルギー一定) と正準集合 (温度一定) です。

具体的な利用方法を説明する前に、一般的な注意点を述べます。まず、分子動力学シミュレーションを行うと `file_names.data` ファイルの識別子 `F_ENF` で指定されるファイルに各ステップでの運動エネルギーがハートリー単位で記述されますが、運動エネルギーと温度との間には  $E_{\text{kin}} = \frac{3}{2} \times N_{\text{atom}} \times k_{\text{B}}T$  という関係があります。ここで運動エネルギーを  $E_{\text{kin}}$ 、原子数を  $N_{\text{atom}}$ 、ボルツマン定数を  $k_{\text{B}}$ 、瞬間的な温度を  $T$  と記述しました。従って運動エネルギーから系の温度を知りたい場合、まず運動エネルギーを原子数で割り、 $3.1578 \times 10^5$  という値を掛けて (ハートリー単位から  $k_{\text{B}}T$  単位への変換)、さらに  $\frac{3}{2}$  で割って下さい。また、原子座標の更新回数は入力で指定した時間刻みの値 (`structure_evolution` ブロック内の `dt` という変数で指定) を掛けることによって実時間で経過が分かります。時間の単位は入力で指定することが可能ですが、原子単位系を利用した場合 (デフォルト)  $2.418 \times 10^{-17}$  という値を掛ければ「秒」に変換することが可能です。たとえば 100 a.u. という時間は 2.418 fs に相当します。

### 9.2 小正準集合

PHASE は、速度ベルレ法を利用してニュートンの運動方程式を時間発展させることによって小正準集合の分子動力学シミュレーションを行います。本節ではこの機能について説明します。

#### 9.2.1 入力の説明

本節で説明する分子動力学シミュレーションのサンプルが、サンプルディレクトリーの `molecular_dynamics/NVE` にあるのでこの入力を利用して説明します。

まず、入力ファイルを示します。

```
accuracy{
  cutoff_wf = 9.00 rydberg
  cutoff_cd = 36.00 rydberg
  num_bands = 8
  xctype = ldapw91
  force_convergence{
    max_force = 1.0e-8 Hartree/Bohr
  }
  initial_wavefunctions = matrix_diagon
  ksampling{
    mesh{
      nx = 4
      ny = 4
      nz = 4
    }
  }
  scf_convergence{
    delta_total_energy = 1e-12 Hartree
    succession = 3
  }
}
...
...
structure{
  unit_cell_type = primitive
  unit_cell{
```

```

        a_vector = 0.0000000000      5.1300000000      5.1300000000
        b_vector = 5.1300000000      0.0000000000      5.1300000000
        c_vector = 5.1300000000      5.1300000000      0.0000000000
    }
    atom_list{
        atoms{
            #tag element rx ry rz mobile
            Si 0.130 0.130 0.130 yes
            Si -0.130 -0.130 -0.130 yes
        }
    }
    element_list{
        #tag element atomicnumber
        Si 14
    }
}
...
...
structure_evolution{
    method = velocity_verlet
    dt = 100
}
...
...
```

この入力、シリコン結晶の入力を少し変更したものとなっています。structure.atom\_list.atoms では、各原子の“mobile”変数を“yes”と設定しています。ここを“no”あるいは“0”と設定すると、その原子は分子動力学シミュレーションを行っても動くことはありません。さらに座標値をあえて安定でない値にしています。具体的には、Si 結晶の二つの原子を (111) 方向にお互いから離れるように少しだけずらしています。

structure\_evolution ブロックでは、“method”変数を“velocity\_verlet”としています。この選択によって小正準集合の分子動力学シミュレーションを行うことができます。また、各ステップでの更新量 (変数 dt) を、原子単位で“100”としています。上で述べたように、この値は  $2.418 \times 10^{-15} \text{s}$  に相当します。

ここまで説明したサンプルの入力を利用すると、原子の初期速度は全て 0 と設定されます。原子に初期速度を与える場合、下記のような入力を準備してください。

```

structure_evolution{
    method = velocity_verlet
    dt = 100
    temperature_control{
        thermostat{
            #tag temp
            300
        }
    }
}
}
```

ここで、“temp”変数で初期の温度をケルビン単位で設定します。原子の初期速度は、この温度になるように、かつ正規乱数に従って、全運動量が 0 になるように設定されます。

原子ごとに異なる初期温度を設定することも可能です。この場合、まず下記のような入力を作成します。

```

structure_evolution{
    method = velocity_verlet
    dt = 100
    temperature_control{
        thermostat{!#tag temp
                    300
                    500

```

```

    }
  }
}

```

ついで、下記のように structure.atom\_list.atoms テーブルの各原子に、“thermo\_group” という変数を設定します。

```

structure{
  ...
  atom_list{
    atoms{
      !#tag rx ry rz  element mobile weight  thermo_group
      0.1159672611      0.1235205209      0.1215156388      Si   1   1   1
      -0.1329067626     -0.1264216714     -0.1225370484      Si   1   1   2
      0.1273740089      0.6305999369      0.6247606249      Si   1   1   3
      ...
      ...
    }
  }
  ...
}

```

この例では一番目の原子が 300K に、二番目の原子が 500K に、三番目の原子が 700K になるよう初期速度が設定されます。

### 9.2.2 出力結果

分子動力学シミュレーションによって得られる結果を説明します。

**原子座標** 原子座標は、構造緩和を行った場合と同様、file\_names.data 中の F\_DYNM 識別子によって指定されるファイル (既定のファイル名は nfdynm.data) に記述されます。このファイルは付属の perl スクリプト、animate.pl などを利用して BioStationViewer や内蔵の PHASE Viewer などが読み込める形式に変換することも可能となっています。

**各ステップでのエネルギー** この情報は、file\_names.data 中の F\_ENF 識別子によって指定されるファイル (既定のファイル名は nfefn.data) に書き出されます。サンプルによって得られる結果を以下に記します。

iter_ion	iter_total	etotal	ekina	econst	forcmx
1	18	-7.8953179624	0.0000042358	-7.8953179624	0.0186964345
2	30	-7.8953851218	0.0000665502	-7.8953185716	0.0183575424
3	43	-7.8955768901	0.0002565396	-7.8953203505	0.0173392067
4	56	-7.8958649874	0.0005418445	-7.8953231430	0.0156398790
5	69	-7.8962052587	0.0008785990	-7.8953266596	0.0132645441
6	83	-7.8965425397	0.0012120826	-7.8953304571	0.0102355854
7	97	-7.8968179539	0.0014840140	-7.8953339398	0.0066063151
8	111	-7.8969784478	0.0016420281	-7.8953364197	0.0024736141
9	125	-7.8969875377	0.0016502900	-7.8953372478	0.0020111576
10	139	-7.8968352058	0.0014992046	-7.8953360011	0.0066379641
11	153	-7.8965440599	0.0012113794	-7.8953326806	0.0111430822
.....					
.....					
.....					

一列目は原子座標の更新回数、二列目は電子の SCF 計算の回数です。三列目は、系の内部エネルギー、四列目は系の運動エネルギーです。五列目は系の内部エネルギーと運動エネルギーを足した値であり、小正準集合における保存量です。最後の六列目は原子にかかる力の最大値です。参考のため、図 11 にサンプルから得られる内部エネルギー、運動エネルギー、全エネルギーとシミュレーションの時間との関係を図示します。

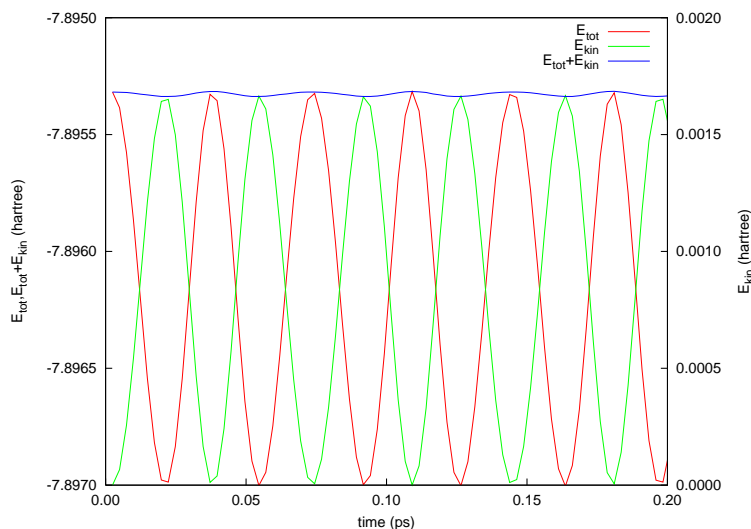


図 11: 内部エネルギー，運動エネルギー，全エネルギーと時間の関係。

### 9.3 正準集合

PHASE は，Nosé-Hoover 熱浴によってイオンの温度を制御しながら分子動力学シミュレーションを行うことによって正準集合の分子動力学シミュレーションを行います。本節ではこの機能について説明します。

#### 9.3.1 入力の説明

本節で説明する分子動力学シミュレーションのサンプルが，サンプルディレクトリーの `molecular_dynamics/NVT` にあるので，この入力を利用して説明します。

この入力では，以下に記すような設定がなされています。

熱浴の設定 `structure_evolution` ブロックに `temperature_control` ブロックを作成する必要があります。設定例を以下に記します。

```
structure_evolution{
  method = temperature_control
  dt      = 50.0
  temperature_control{
    thermostat{
      #tag temp  qmass
      300      5000
    }
  }
}
```

上記の入力例では，まず “method” 変数を `temperature_control` としています。この変数によって PHASE に温度制御を行うよう指示します。ついで，“dt” 変数を設定しています。これは，時間刻みの指定です。原子単位で入力します。例で示されている 50.0 という値は，約 1.2fs に相当します。

さらに，`temperature_control` ブロックで熱浴の詳細設定を行っています。“thermostat” ブロックで各熱浴のパラメータを設定します。“temp” パラメータによってその熱浴の目的とする温度 (ケルビン単位)，“qmass” パラメータによって熱浴の質量 (原子単位) を設定します。

熱浴の割り当て `structure` ブロックの，`atoms` ブロックを設定する必要があります。設定例を以下に記します。

```
structure{
  ...
```

```

atom_list{
  num_atoms = 8
  coordinate_system = internal
  atoms{
    !#tag rx ry rz  element mobile weight  thermo_group
      0.1159672611      0.1235205209      0.1215156388      Si   1   1   1
      -0.1329067626     -0.1264216714     -0.1225370484      Si   1   1   1
      0.1273740089      0.6305999369      0.6247606249      Si   1   1   1
      -0.1152089939     -0.6164829779     -0.6221565128      Si   1   1   1
      0.6299472943      0.1341313888      0.6253193197      Si   1   1   1
      -0.6305720382     -0.1290073650     -0.6187967685      Si   1   1   1
      0.6151271805      0.6206113965      0.1333834419      Si   1   1   1
      -0.6276524003     -0.6268549639     -0.1175099372      Si   1   1   1
  }
}
...
}

```

基本的にはこれまでの設定と同様ですが、各原子に “thermo\_group” パラメータを割り振っていることにご注意ください。このパラメータに先に説明した熱浴の識別番号を設定します。なお、熱浴の識別番号は熱浴の定義順に割り振られます。また、他の属性値と同様、“#default” タグを利用することによってデフォルト値を設定することも可能です。この例では全ての原子に同じ thermo\_group を設定していますが、各原子が異なる熱浴に関連付けられていても問題ありません。

### 9.3.2 熱浴の質量の設定方法

熱浴の質量 (以後、 $Q$  と表記します) は、適切に設定する必要があります。小さすぎる  $Q$  を採用すると、系のダイナミクスに熱浴に起因する人為的なモードが生じてしまい、また場合によっては計算が破綻してしまいます。他方大きすぎる  $Q$  を採用すると、系が熱平衡に達するのに多くのステップ数を必要とするようになってしまいます。

$Q$  の値は、系の特徴的な振動の周期と熱浴の振動の周期がおおよそ同等か、熱浴の方が少し長くなるように選ぶとよいとされています。熱浴の振動の周期と系の振動の周期の関係は、大雑把には次の式で評価できます (S. Nosé, Progress of Theoretical Physics Supplement No 103, 1991, pp.1-46)。

$$\tau = \frac{2\pi}{\omega} = 2\pi \left( \frac{Q}{2gk_B T} \right)^{1/2} \quad (9)$$

ここで  $\tau$  はそれぞれ系の周期と周波数、 $g$  は系の自由度 ( $3 \times$  その熱浴に関連付けられている原子の数)、 $k_B$  はボルツマン定数、 $T$  がその熱浴の温度です。例えば、 $\tau$  を 0.05 ps、原子の数を 8、温度を 300 K として (9) 式で  $Q$  の値を見積もると、原子単位でおおよそ 4600 程度となります。

### 9.3.3 出力の解析

温度制御を行ったシミュレーションは、次のように解析することが可能です。

**原子座標** 原子座標は、構造緩和を行った場合、通常の分子動力学シミュレーションを行った場合と同様、file\_names.data 中の F\_DYNM 識別子によって指定されるファイル (既定のファイル名は nfdynm.data) に記述されます。

**各ステップでのエネルギー** この情報も、構造緩和や通常の分子動力学シミュレーションの場合と同様 file\_names.data 中の F\_ENF 識別子によって指定されるファイル (既定のファイル名は nfenf.data) に書き出されます。内容は第 9.2.2 節で紹介したものと同等です。ただし、F\_ENF ファイルに書き出される全エネルギーには、熱浴の内部エネルギーと運動エネルギーが足されます。正準集合ではこのエネルギーが保存量となります。参考のため、サンプルの入力を利用して得られる、1 ps 程度のシミュレーションを行った際の瞬間的な温度と時間の関係を図 12 に図示します。温度は、全ての原子について 300K としています。



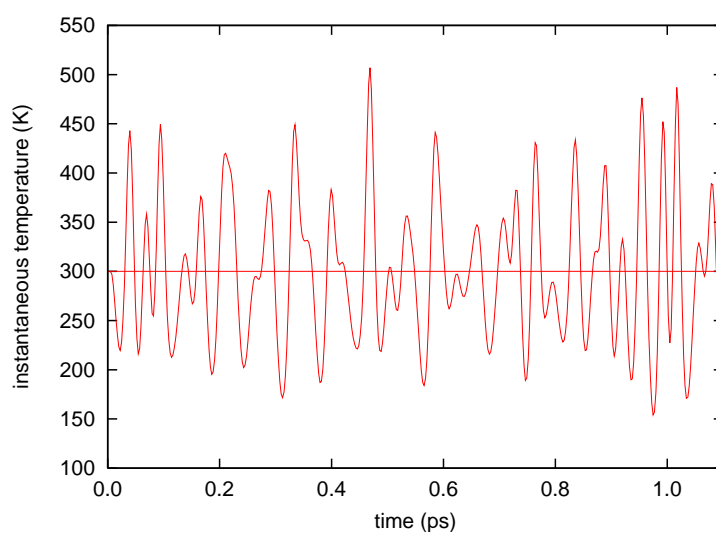


図 12: 瞬間的な温度と時間の関係。

## 10 振動解析

### 10.1 はじめに

PHASE には格子振動の規準モードを計算する機能が組み込まれています。本章ではその機能について説明します。本機能を有効にすると、まず、原子を平衡位置からわずかに変位させて力計算を行われます。そのあとで、その力のデータから力定数行列が求められ、それから動力学行列が構築されます。その動力学行列の固有値問題が解かれて、規準振動の振動数と固有ベクトルが求められます。

### 10.2 振動解析の理論

$i$  番目の原子の安定位置から変位を  $\mathbf{u}_i$  とします。変位が微小で二次以上の項が無視できるとき、格子系の運動方程式は

$$m_i \ddot{u}_{i\alpha} = - \sum_{j\beta} \Phi_{i\alpha,j\beta} u_{j\beta} \quad (10)$$

と書けます。 $\Phi_{i\alpha,j\beta}$  は力の定数で、原子変位に関する系のエネルギー  $E(\mathbf{u}_1, \mathbf{u}_2, \dots)$  の二階微分として定義されています。

$$\Phi_{i\alpha,j\beta} = \frac{\partial^2 E}{\partial u_{i\alpha} \partial u_{j\beta}} \quad (11)$$

力の定数はヘルマン-フラインマン力を原子変位で微分することにより、求めることができます。

$$\Phi_{i\alpha,j\beta} = - \frac{\partial F_{i\alpha}}{\partial u_{j\beta}} \quad (12)$$

本プログラムでは、この微分は中央差分近似で行われます。変位パラメーターを  $a$  とすると

$$\frac{\partial F_{i\alpha}}{\partial u_{j\beta}} = \frac{F_{i\alpha}|_{u_{j\beta}=a} - F_{i\alpha}|_{u_{j\beta}=-a}}{2a} \quad (13)$$

と書けます。力の定数には結晶の対称性による制約があり、これを満たすように力の定数を補正する必要があります。第一に、 $i, j$  原子が空間群の対称操作  $\{R | \mathbf{T}\}$  で  $i', j'$  原子に移るとき、力定数テンソル  $\Phi_{i,j}$  は力定数テンソル  $\Phi_{i',j'}$  を回転行列  $R$  で回転させたものに等しいです。つまり、

$$R\mathbf{r}_i + \mathbf{T} = \mathbf{r}_{i'} \quad (14)$$

$$R\mathbf{r}_j + \mathbf{T} = \mathbf{r}_{j'} \quad (15)$$

ならば、

$$\Phi_{i,j} = R^T \Phi_{i',j'} R \quad (16)$$

でなければなりません。第二に、力定数テンソル  $\Phi_{i,j}$  の成分  $\alpha\beta$  を原子の番号  $j$  すべてにわたり足し合わせると、ゼロになります。つまり、

$$\sum_j \Phi_{i\alpha,j\beta} = 0 \quad (17)$$

です。第三に、力定数行列は対称でなければなりません。つまり、

$$\Phi_{i\alpha,j\beta} = \Phi_{j\beta,i\alpha} \quad (18)$$

です。換算変位  $w_{i\alpha} = u_{i\alpha} / \sqrt{m_i}$  と動力学行列  $D_{i\alpha,j\beta} = \Phi_{i\alpha,j\beta} / \sqrt{m_i m_j}$  を用いて、格子系の運動方程式 (10) を

$$\ddot{w}_{i\alpha} = - \sum_{j\beta} D_{i\alpha,j\beta} w_{j\beta} \quad (19)$$

と書き換えます。この方程式を解くために、 $w_{i\alpha} = Q \xi_{i\alpha} e^{i\omega t + \delta}$  という解を仮定します。

$$\omega^2 \xi_{i\alpha} = \sum_{j\beta} D_{i\alpha,j\beta} \xi_{j\beta} \quad (20)$$

これは固有値が  $\omega^2$  で、固有ベクトルが  $\xi_{i\alpha}$  となる、行列  $D_{i\alpha,j\beta}$  の固有値問題です。PHASE の振動解析機能ではこの固有値問題を解き、格子振動の規準モードを求めます。

### 10.3 入力の説明

振動解析の入力変数を表 8 にあげます。以下で、各変数の説明をします。

#### 10.3.1 原子座標と対称性の入力

原子座標は反転対称があってもすべて入力する必要があります。そのため、`sw_inversion` は OFF に設定します。振動モードの分類と入力座標の対称性チェックに系の空間群を使用するので、その空間群を `symmetry` ブロックで正しく指定します。

#### 10.3.2 元素の質量の指定

元素の質量は `element_list` ブロックの変数 `mass` で指定する。原子単位 (a.u.) ではなく、原子質量単位 (amu) で入力するには、`#units atomic_mass` を `#tag` 行の上に挿入する。

#### 10.3.3 原子変位の選択

原子変位は `Phonon` ブロックの `displacement` で設定します。通常、原子変位は 0.1 a.u. 以下にとると良いです。振動数の原子変位依存性を調べて、希望する振動数の収束が得られる原子変位に設定します。`norder` を 2 に設定することで、差分の次数を 3 から 5 に換えることができます。`displacement` で設定した値を `u` とすれば、原子変位は `-u, -u/2, u/2, u` になります。`sw_polynomial_fit` を ON にして多項式フィットにすれば、`norder` を 2 より大きく設定できます。そのときの原子変位は `-u/norder, -u/(norder-1), ..., u/(norder-1), u/norder` です。`norder` を大きくすると微分精度はよくなりますが、力計算の回数が増えるので、計算時間が `norder` が 1 の場合の  $2^{*norder}$  倍になるので注意してください。

### 10.4 実行の方法

振動解析を行うには、まず原子を平衡位置に置かなければなりません。これは、構造最適化機能を用いて行います。構造最適化の方法については 8.5 節を参照してください。構造最適化計算が終了したら、`nfdynm.data` の最後に記述されている最適構造での PHASE の入力を作成します。これに、以下のような `Phonon` ブロックを加え、振動解析の設定を行います。

```
Phonon{
  sw_phonon = on
  sw_calc_force = on
  sw_vibrational_modes = on
  displacement = 0.05
}
```

構造最適化計算と同じように擬ポテンシャルファイル、`file_names.data` を用意して PHASE を実行します。計算が終了すると、振動解析の結果 `mode.data` が出力されます。また、計算した力のデータが `force.data` に出力されます。

### 10.5 振動解析結果の出力ファイル mode.data

`mode.data` には振動解析の結果が記述されます。まず最初に基本並進ベクトル  $\mathbf{a}_i = (a_{ix}, a_{iy}, a_{iz})$  が次の形式で記述されます。

```
--- primitive lattice vectors ---
a_1x a_1y a_1z
a_2x a_2y a_2z
a_3x a_3y a_3z
```

次に原子の数 `natm` と各原子の座標  $(x_i, y_i, z_i)$  と質量  $m_i$  とラベル `name(i)` が次の形式で記述されます。

```

--- Equilibrium position and mass of each atom---
Natom = natm
do i=1,natm
  i  x(i)  y(i) z(i) m(i) name(i)
end do

```

次に振動解析の結果が次の形式で記述されます。

```

--- Vibrational modes ---
Nmode= nmode  Natom= natm
do m = 1,nmode
  n=  m  representation(m) acvtive(m)
  hbarW= omega_ha(m) Ha = omega_ev(m) eV; nu= omega_nu(m) cm^-1
  do i=1,natm
    i  vec(m,i,1) vec(m,i,2) vec(m,i,3)
  end do
end do

```

representation は既約表現の配列です。active(m) はラマン活性なモードあれば R になり、赤外活性なモードであれば IR となります。両活性であれば、IR&R となります。サイレントモードの場合には何も表示されません。vec は固有ベクトルの配列で、omega\_ha は Hartree 単位での振動数で、omega\_ev は電子ボルト単位での振動数で、omega\_nu は波数です。

## 10.6 カデータファイル force.data の入出力

force.data には力の定数を計算するための力のデータが記述されます。その力データは次の形式で出力されます。

```

num_force_data, norder, sw_ploynomial_fit
do i = 1, num_force_data
  displaced_atom, displacement(1:3)
  do ia = 1, natm
    i, force_data(ia,1:3,i)
  end do
end do

```

num\_force\_data は力を計算する原子配置の数で、displaced\_atom は変位した原子の番号で、配列 displacement が原子の変位ベクトル ( $u_x, u_y, u_z$ ) です。norder は入力で指定した norder の値が記述されます。sw\_ploynomial\_fit は入力の sw\_ploynomial\_fit が ON のときに、ON を表す 1 が記述されます。OFF の場合には、0 が記述されます。

sw\_calc\_force を OFF に設定することで、出力された力を読み込み、振動解析をやり直すことができます。元素の質量を変更することは問題ありませんが、力計算に関連する変数は変更してはなりません。

## 10.7 例：水分子の振動解析

### 10.7.1 構造最適化

振動解析を行うには原子が平衡状態になければなりませんので、振動解析を行うときと同じ条件で構造最適化を行います。平衡状態ないと動力学行列の固有値が負になり、振動数が純虚数のソフトモードが現れます。水分子の構造最適化の入力例を以下に示します。

```

Control{
  condition = initial
  cpumax = 1 day ! maximum cpu time
  max_iteration = 6000
}

accuracy{
  cutoff_wf = 25.00 rydberg
}

```

```

cutoff_cd = 225.00 rydberg
num_bands = 8
xctype = ggapbe
initial_wavefunctions = matrix_diagon
matrix_diagon {
    cutoff_wf = 5.0 rydberg
}
ksampling{
    method = gamma
}
scf_convergence{
    delta_total_energy = 1.e-10
    succession = 3
    num_max_iteration = 300
}
force_convergence{
    delta_force = 1.e-4
}
initial_charge_density = Gauss
}

structure{
    unit_cell_type = primitive
    unit_cell{
        a_vector = 15.0          0.0          0.0
        b_vector = 0.0          15.0          0.0
        c_vector = 0.0          0.0          15.0
    }
    symmetry{
        tspace{
            lattice_system = primitive
            generators{
                #tag rotation tx ty tz
                C2z      0 0 0
                IC2x     0 0 0
            }
        }
        sw_inversion = off
    }
}

magnetic_state = para

atom_list{
    coordinate_system = cartesian
    atoms{
        !#default mobile=on
        !#tag rx ry rz element
        -1.45 0.000 1.123 H
        1.45 0.000 1.123 H
        0.0 0.0 0.0 O
    }
}

element_list{ #units atomic_mass
    #tag element atomicnumber zeta dev
    H 1 1.00 0.5
    O 8 0.17 1.0 }

```

```

}

wf_solver{
  solvers {
    !#tag sol      till_n dts dte itr  var      prec cmix submat
      msd         5      0.1 0.1  1    tanh on   1    on
      lm+msd      10      0.1 0.4  50    tanh on   1    on
      rmm2p       -1      0.4 0.4  1    tanh on   2    on
  }
  rmm {
    edelta_change_to_rmm = 1.d-6
  }
  lineminimization {
    dt_lower_critical = 0.1
    dt_upper_critical = 3.0
  }
}

charge_mixing{
  mixing_methods {
    !#tag id method  rmxs rmxe itr var      prec istr nbxmix update
      1  broyden2  0.3  0.3  1   linear on   5   10   RENEW
      2  simple   0.2  0.5  100 linear on   *   *    *
  }
}

structure_evolution{
  method = cg
}

```

file\_names.data には element\_list と同じ順番でポテンシャルファイル H\_ggapbe.nc.01.pp と O\_ggapbe.us.02.pp を指定します。この入力を使用して得た水分子の構造を図 13 に示します。

### 10.7.2 振動解析

構造最適化後に振動解析を行うには、入力の原子座標を最適化した座標に換えて、Phonon ブロックを加え、振動解析の設定をします。最適化原子座標は構造最適化計算の出力ファイル nfdynm.data に記述されている最後のステップの原子座標です。

```

atom_list{
  coordinate_system = cartesian
  atoms{
    !#tag rx          ry          rz          element
      -1.446816228    0.000      1.123327795 H
      1.446816228    0.000      1.123327795 H
      0.0             0.0        0.0          O
  }
}

```

振動解析の設定はたとえば以下のようにします。

```

Phonon{
  sw_phonon = on
  sw_calc_force = on
  sw_vibrational_modes = on
  displacement = 0.05
}

```

原子変位は 0.05 とします。この入力を使用して得た水分子の規準モードの振動数と固有ベクトルをそれぞれ図 14 と図 15 に示します。振動数レベル図は Perl スクリプト freq.pl を使用して作成できます。チュートリアルに使用法が書かれていますが、分子の場合には以下のように -mol というオプションを付けて freq.pl を実行します。

```
$ freq.pl -mol mode.data
```

規準振動の固有ベクトルの図は BioStationViewer を使用して作成できます。control.inp に原点の移動量を以下のように記述します。

```
origin 7.5 7.5 7.5
```

この control.inp をコマンド引数に追加して、animate.pl を以下のように実行します。

```
$ animate.pl mode.data control.inp
```

こうして生成される mode\_7.tr2, mode\_8.tr2, mode\_9.tr2 が図示した振動モードの拡張 trajectory 形式のファイルです。

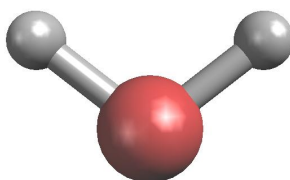


図 13: 水分子の構造

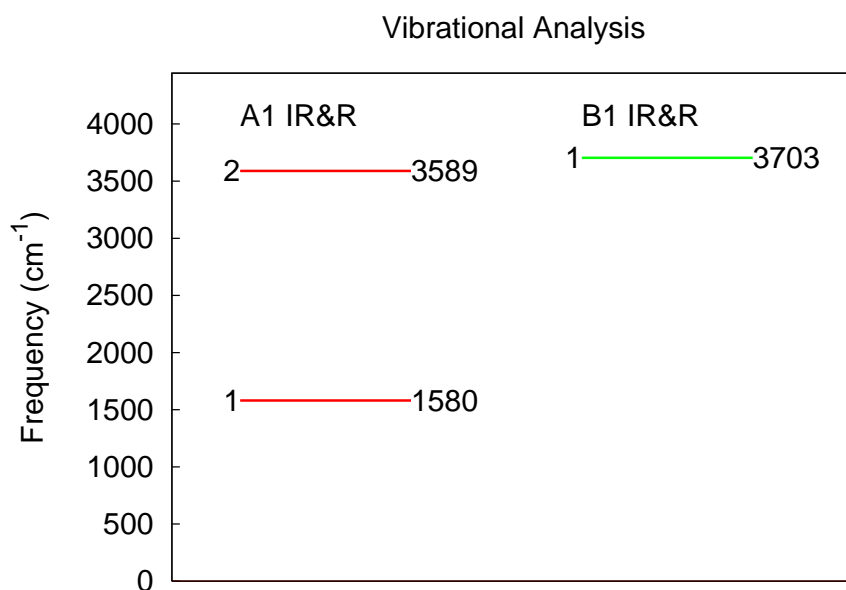


図 14: 水分子の振動モードの振動数

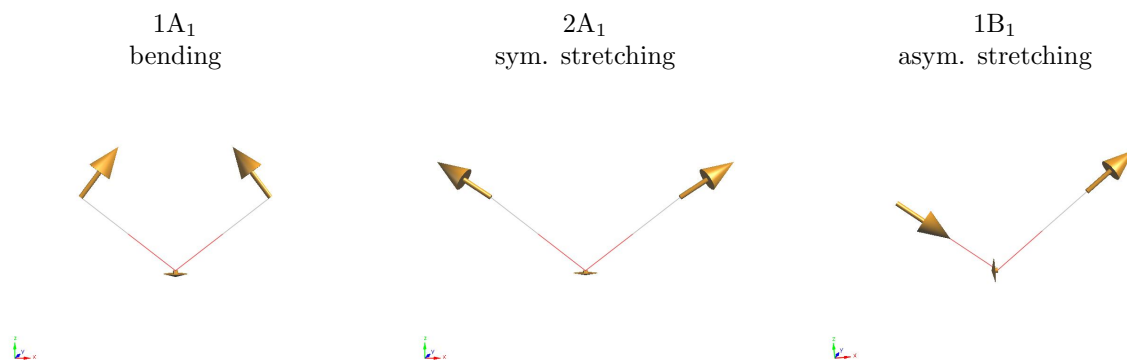


図 15: 水分子の振動モードの固有ベクトル

変数名またはタグ名	既定値	説明
sw_phonon	OFF	格子振動解析設定ブロックを有効にするかどうかのスイッチです。
sw_calc_force	OFF	振動解析のための力計算を行うかどうかのスイッチです。ON のときには、格子振動解析のための力計算を行います。(計算した力は force.data に出力されます。)OFF のときには、sw_vibrational_modes=ON ならファイル"FORCE"から力のデータを読み込みます。
displacement	0.1	原子変位パラメーター。
sw_vibrational_modes	OFF	格子振動解析を行うかどうかのスイッチです。ON のときには、格子振動解析が行われ、modes.data に結果が出力されます。OFF のときには、格子振動解析は行われません。
norder	1	差分次数を変更するパラメーターです。
sw_polynomial_fit	OFF	差分ではなく、多項式フィットで力の微分を求めるときに ON にします。

表 8: 振動解析に関する変数の説明



## 11 STM 像のシミュレーションプログラム

ここで説明するプログラム「STM」が対象とする系は、シリコン原子 (Si) などからなるスラブ層と真空領域から構成される系です。例えば、Si 層の表面に吸着した水素 (H) や酸素 (O) の STM 像をシミュレートするのに、10 原子層程度の厚さの Si 層 (スラブ層) と各スラブ層間に電子移動が生じない程度の十分な厚さの真空層を交互に積み重ねた模型 (単位格子には Si の層と真空層がひとつずつ含まれる) を用います。この Si 層の表面に H や O を吸着させた構造を用意し、構造最適化を行い、収束させた電子状態から STM 像をシミュレートし実験と対比するのです。バイアス条件に対応したエネルギー窓 (フェルミエネルギーレベルを  $E_F$  とすると、例えば、 $E_F - 2$  (eV) から  $E_F$  まで) 内にある電子状態密度の表面から 1 ~ 2 離れた位置における断面図をもって STM 像をシミュレートしたとすることもできますが、こうした STM 像は実験の像とかなり異なったものになっている可能性もあります。実際の STM 探針は、表面からかなり (~ 5 以上) 離れているからです。STM 探針のある位置での電子状態密度分布、あるいはその高さあたりの電子状態密度一定となる高さを濃淡に変えたものの方が実際の STM 像をよりよくシミュレートするものになるはずですが。

しかし、第一原理手法を使ったプログラム PHASE で表面から離れた真空領域の電子状態を計算してみると、かなり乱れた状態 (ノイズの入ったような状態) になっていることが多いです。これは、真空領域では電子状態密度がバルク中の値の何桁も小さいものになっていて、平面波基底では精度よく求めるのに限界があるからです (真空領域においてこのような限界があっても、PHASE の全エネルギー計算に関する誤差は小さいです)。そこで、PHASE で計算したポテンシャルの下で、Schrödinger 方程式を解き、真空層領域の波動関数を精度よく求めることにします。解くべき Schrödinger 方程式に関しては、影島博之氏の博士論文 [1] の内容を引用して説明します。具体的な解法は、加藤弘一氏 [2] によります。ここで説明する方法で解いた波動関数を使って、より精度のよい STM 像を作ることができます。

### 11.1 計算手法

解くべき方程式

対象物の表面 (から適当な距離離れた位置) から真空層の中心までの領域で、一次元の Schrödinger 方程式を解きます。このとき、この領域の両端 (表面側と真空の中心側) とで、PHASE で得た波動関数と一致するという境界条件を課します。影島博之氏の博士論文 [1] 中にある式を書き下すと次のようになります。

$$E_{\mu, \mathbf{k}_{\parallel}} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z) = \frac{\hbar^2}{2m} (\mathbf{k}_{\parallel} + \mathbf{G}_{\parallel})^2 \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z) - \frac{\hbar^2}{2m} \frac{\partial^2}{\partial z^2} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z) + \sum_{\mathbf{G}'_{\parallel}} V(\mathbf{G}_{\parallel} - \mathbf{G}'_{\parallel}, z) \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}'_{\parallel}, z) \quad (21)$$

ここで、

$$\Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{r}) = \sum_{\mathbf{G}_{\parallel}} e^{i(\mathbf{k}_{\parallel} + \mathbf{G}_{\parallel}) \cdot \mathbf{r}_{\parallel}} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z), \quad (22)$$

$$V(\mathbf{r}) = \sum_{\mathbf{G}_{\parallel}} e^{i\mathbf{G}_{\parallel} \cdot \mathbf{r}_{\parallel}} V(\mathbf{G}_{\parallel}, z). \quad (23)$$

です。有効ポテンシャル  $V(\mathbf{r})$  は、第一原理計算プログラム PHASE で得られた遮蔽ポテンシャルを使います。但し、本来の  $V(\mathbf{r})$  でなく非局所項を取り去ったもの、すなわち局所ポテンシャルのみ、を使うことにします。

フーリエ変換関係

ここで、波動関数  $\Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{r})$  を次のように項を分けて

$$\Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{r}) = e^{i\mathbf{k}_{\parallel} \cdot \mathbf{r}_{\parallel}} \Phi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{r}) \quad (24)$$

とおくと、

$$\Phi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{r}) = \sum_{\mathbf{G}_{\parallel}} e^{i\mathbf{G}_{\parallel} \cdot \mathbf{r}_{\parallel}} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z) \quad (25)$$

$$\Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z) = \frac{1}{\sum_{m,n}} \sum_{m,n} e^{-i\mathbf{G}_{\parallel} \cdot (x_m, y_n)} \Phi_{\mu, \mathbf{k}_{\parallel}}((x_m, y_n), z) \quad (26)$$

のフーリエ変換の関係が導けます。 $m, n$  は  $x$ - $y$  面内の FFT 格子点の位置を示す添字です。

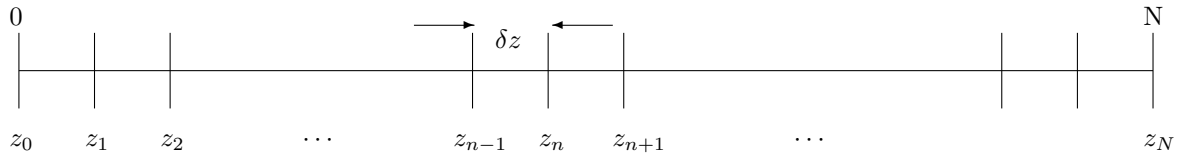
### ポテンシャルと波動関数の畳み込み

(21) 式の右辺第 3 項のポテンシャルと波動関数の畳み込みの項は、次の二次元 FFT を用いて計算できます。

$$\sum_{\mathbf{G}'_{\parallel}} V(\mathbf{G}_{\parallel} - \mathbf{G}'_{\parallel}, z) \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}'_{\parallel}, z) = \frac{1}{\sum_{m,n}} \sum_{m,n} e^{-i\mathbf{G}_{\parallel} \cdot (x_m, y_n)} V(x_m, y_n, z) \Phi_{\mu, \mathbf{k}_{\parallel}}(x_m, y_n, z) \quad (27)$$

### 差分方程式化

真空層領域を  $z$  軸に関して  $z_0$  から  $z_N$  まで  $N$  等分します。 $z_0$  は表面側、 $z_N$  は真空側です。



そうして、(21) 式を次のように差分化します。

$$\begin{aligned} E_{\mu, \mathbf{k}_{\parallel}} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_n) &= |\mathbf{k}_{\parallel} + \mathbf{G}_{\parallel}|^2 - \frac{1}{\delta z^2} \{ \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_{n+1}) - 2\Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_n) + \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_{n-1}) \} \\ &\quad + V(0, z_n) \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}, z_n) + \sum_{\mathbf{G}' \neq \mathbf{G}} V(\mathbf{G}_{\parallel} - \mathbf{G}'_{\parallel}, z) \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}'_{\parallel}, z) \end{aligned} \quad (28)$$

これをさらに書き換えて、

$$\begin{aligned} \delta z^{-2} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_{n-1}) &+ \{ E_{\mu, \mathbf{k}_{\parallel}} - |\mathbf{k}_{\parallel} + \mathbf{G}_{\parallel}|^2 - 2\delta z^{-2} - V(0, z_n) \} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_n) + \delta z^{-2} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_{n+1}) \\ &= \sum_{\mathbf{G}' \neq \mathbf{G}} V(\mathbf{G}'_{\parallel} - \mathbf{G}_{\parallel}, z) \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}'_{\parallel}, z_n) \end{aligned} \quad (29)$$

を得ます。但し、次の境界条件が課されています。

$$\begin{cases} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_0) = \Psi_{\mu, \mathbf{k}_{\parallel}}^0(\mathbf{G}_{\parallel}, z_0) \\ \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_N) = \Psi_{\mu, \mathbf{k}_{\parallel}}^0(\mathbf{G}_{\parallel}, z_N) \end{cases} \quad (30)$$

### 解法

(29) 式は次の 3 重対角行列を使って表すことができます。

$$\begin{pmatrix} a_{11} & a_{12} & 0 & & & & & & \\ a_{21} & a_{22} & a_{23} & & & & & & \\ 0 & a_{32} & a_{33} & a_{34} & & & & & \\ & & & \ddots & \ddots & & & & \\ & & & & \ddots & & & & \\ & & & & & \ddots & & & \\ & & & & & & a_{N-2,N-3} & a_{N-2,N-2} & a_{N-2,N-1} & 0 \\ & & & & & & a_{N-1,N-2} & a_{N-1,N-1} & a_{N-1,N} & \\ & & & & & & 0 & a_{N,N-1} & a_{N,N} & \end{pmatrix} \begin{pmatrix} \Psi(z_1) \\ \vdots \\ \Psi(z_i) \\ \vdots \\ \Psi(z_N) \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_i \\ \vdots \\ c_N \end{pmatrix} \quad (31)$$

ここでは、表式の単純化のために、 $\Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_i)$  を  $\Psi(z_i)$  と表しました。 $a_{ij}$ ,  $c_i$  は次の式のとおりです。

$$\begin{cases} a_{ii} = E_{\mu, \mathbf{k}_{\parallel}} - |\mathbf{k}_{\parallel} + \mathbf{G}_{\parallel}|^2 - 2\delta z^{-2} - V(0, z_i) \\ a_{i,i-1} = a_{i-1,i} = \delta z^{-2} \\ c_i = \sum_{\mathbf{G}' \neq \mathbf{G}} V(\mathbf{G} - \mathbf{G}', z_i) \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}', z_i) \\ c_1 = \sum_{\mathbf{G}' \neq \mathbf{G}} V(\mathbf{G} - \mathbf{G}', z_1) \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}', z_1) - \delta z^{-2} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_0) \\ c_N = \sum_{\mathbf{G}' \neq \mathbf{G}} V(\mathbf{G} - \mathbf{G}', z_N) \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}', z_N) - \delta z^{-2} \Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_{N+1}) \end{cases} \quad (32)$$

(32) 式の  $c_N$  の表式に現われる  $\Psi_{\mu, \mathbf{k}_{\parallel}}(\mathbf{G}_{\parallel}, z_{N+1})$  を 0 とおくことにします。(31) 式を  $c_i$  を固定して  $\Psi(z_i)$  に関して解きます。 $c_i$  は  $\Psi$  を含みます。これと、解いた  $\Psi$  とは自己無撞着でなければなりません。そこで、(31) 式を iterative に解くことにします。 $\Psi^m(z_i)$  が与えられれば、 $c_i$  が計算できます。この  $c_i$  のもとで、(31) 式を解き、得られたものを  $\Psi^{m+1}(z_i)$  とします。 $\Psi^m(z_i) - \Psi^{m+1}(z_i)$  のノルムがある設定値以下であれば、収束したとみなし、設定値以上であれば、

$$\Psi^{m+1}(z_i) = r^{m+1} \times \Psi^{m+1}(z_i) + (1 - r^{m+1}) \times \Psi^m(z_i) \quad (33)$$

によって、次の入力  $\Psi(z_i)$  を決定し、再び  $c_i$  を計算するところから始めます。

(31) 式を LU 分解を利用して解くことにします。

$$\begin{aligned} A_{i,j} &= a_{ij}, \\ X_i &= x_i = \Psi(z_i), \\ C_i &= c_i \end{aligned}$$

と置くと、(31) 式は、

$$AX = C \quad (34)$$

と書けます。 $A$  は下三角行列  $L$  と対角成分が 1 の上三角行列  $U$  の積に分解できます。上三角行列  $U$  は、対角成分と  $\{k, k+1\}$  成分以外は全て 0 です。 $u_{k,k+1} = w_k$  とおくと、

$$U = \begin{pmatrix} 1 & w_1 & & & \\ 0 & 1 & w_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & w_{N-1} \\ 0 & & & & 1 \end{pmatrix} \quad (35)$$

です。一方、下三角行列  $L$  は、

$$L = \begin{pmatrix} a_{12}/w_1 & 0 & & & \\ a_{21} & a_{23}/w_2 & & & \\ & & \ddots & & \\ & & & a_{N-1,N-2} & a_{N-1,N}/w_{N-1} \\ & 0 & & a_{N,N-1} & a_{N,N} - a_{N,N-1}w_{N-1} \end{pmatrix} \quad (36)$$

と表されることがすぐに分かります。 $w_k$  は、 $A = LU$  より、

$$w_1 = \frac{a_{12}}{a_{11}} \quad (37)$$

$$w_k = \frac{a_{k,k+1}}{a_{k,k} - a_{k,k-1}w_{k-1}} \quad k = 2, 3, \dots, N \quad (38)$$

の関係が得られ、帰納的に解くことができます。 $AX = LUX = C$  は、

$$LG = C \quad (39)$$

と、

$$UX = G \quad (40)$$

の2つに分けて解くことができます。それぞれ前進代入と後退代入を行います。前進代入部分は、次のとおりです。

$$\begin{aligned} g_1 &= c_1 \frac{w_1}{a_{12}} = \frac{c_1}{a_{11}} \\ g_k &= (c_k - a_{k,k-1}g_{k-1}) \frac{w_k}{a_{k,k+1}} \quad (k = 2, 3, \dots, N) \end{aligned} \quad (41)$$

後退代入部分は、

$$\begin{aligned} x_N &= g_N \\ x_k &= g_k - w_k x_{k+1} \quad (k = N-1, N-2, \dots, 1) \end{aligned} \quad (42)$$

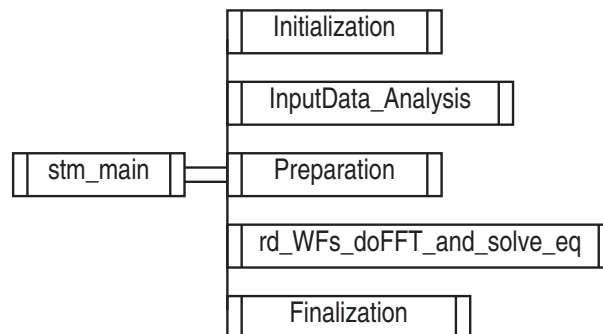
です。

## 11.2 プログラムについて

### 11.2.1 ソースプログラム構造

main ルーチンは次のとおりです。

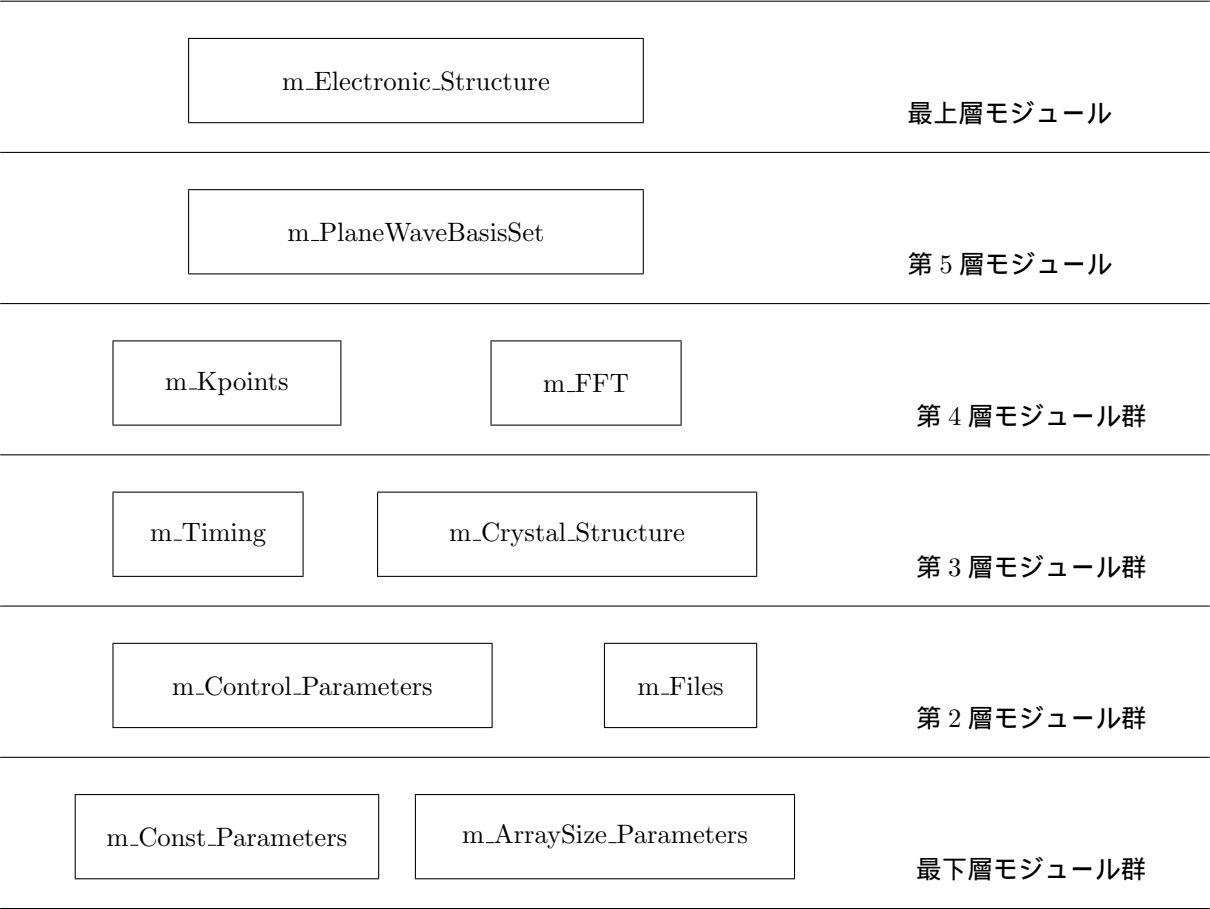
```
program STM
  call Initialization
  call InputData_Analysis
  call Preparation
  call rd_WFs_doFFT_and_solve_eq
  call Finalization
end program STM
```



ここで呼び出されている Initialization, rd\_WFs\_doFFT\_and\_solve\_eq などのサブルーチンを第一層サブルーチンと呼ぶことにします。第一層サブルーチンは、それぞれ必要な処理をモジュールに依頼します。モジュール群は階層化されています。モジュール内では、さらに下位のモジュールに処理を依頼したり、外部サブルーチン呼び出ししたりします。

#### モジュール階層

モジュールの階層は次のようになっています。上層のモジュールは下層のモジュールを（use 文を用いて）使うことができます。



11.2.2 入出力ファイル

入出力ファイルには、次の表に示す 9 つのものがあ

ファイル指示名	番号	既定ファイル名	入出力	内容
F_INP	31	nfinp.data	入力	STM 像のエネルギー範囲などの指示
F_CNTN_BIN	55	continue_bin_stm.data	入力	固有値, 基底逆格子点の集合などのデータ
F_ZAJ	44	zaj.data	入力	波動関数の平面波展開係数
F_VLC	45	vlc.data	入力	ポテンシャルの局所成分
F_CHGU	60	nfchgu.data	出力	Schrödinger 方程式を解いて得た電子状態密度
F_CHGD	61	nfchgd.data	出力	F_CHGU と同様。但し, down spin 状態
F_CHGU_P	46	nfchgu_p.data	出力	入力波動関数から得た電子状態密度
F_CHGD_P	47	nfchgd_p.data	出力	F_CHGU_P と同様。但し, down spin 状態
F_file_names_data	5	stm_file_names.data	入力	上の 8 つのファイルを指定するためのファイル

尚, STM は Gaussian cube 形式の電子状態密度を出力します。

F\_INP

このファイルは STM 像シミュレーションの制御のために用意するテキストファイルです。記入例を示します。  
尚, 長さの単位は Bohr, エネルギーの単位は eV です。

4 1 : kv3, nspin

```

4 1 : kv3,  nspin
  1  0.0000 -0.1624 -0.3247      0.0000 -0.3750 -0.3750
  2  0.0000 -0.1624 -0.1082      0.0000 -0.3750 -0.1250
  3  0.0000 -0.0541 -0.3247      0.0000 -0.1250 -0.3750
  4  0.0000 -0.0541 -0.1082      0.0000 -0.1250 -0.1250
110 42.9043654360 ! neg,  z1 (=lattice c)
-3.00 -2.000 ! e1,  e2 : Efermi+e1 - Efermi+e2
90 ! nlpf
53 90 !izi,  izf
  1 ! colomn number of the z-component
0.10 0.15 100 ! rini,  rfin,  nfin
0.01 ! erlmt

```

1 行目の最初の数字は,  $k$  点の個数です。あとの数字は, スピンの違いを考慮するかしないかを示す数で, 考慮しない場合が 1, する場合が 2 です。2 行目から 4 行目までは  $k$  点の座標です。全部で  $4 \times 1 = 4$  組の座標があります。ここでは, スピンの違いを考慮していないためこうなったが, スピンの違いがある場合の記すべき座標の組の個数は, 1 行目に記した  $k$  点の個数の倍に等しくなります。座標の番号, 直交座標系での座標, 基本逆格子ベクトル系での座標で一組です。6 行目

```
110 42.9043654360 ! neg,  z1 (=lattice c)
```

の最初の数字はエネルギーバンドの数, あとの数字は表面に垂直方向の結晶格子の長さです。7 行目の 2 つの数字は, エネルギー範囲。フェルミエネルギーレベルを基準にします。8 行目の数字は, FFT の格子点の表面垂直方向の数に関係した数です。反転対称がない系では,  $c$  軸方向の FFT 格子点の数そのもの, 反転対称がある系では, その半分の数を指定します。9 行目の 2 つの数字は, 真空領域の Schrödinger 方程式を解く範囲の指定。FFT の格子点でその範囲を指定します。10 行目の数は, 表面垂直方向が  $k$  点の座標のどの成分に対応しているかを指示します。表面垂直軸が第 1 格子ベクトルの方向であれば 1 を, 第 2 格子ベクトルの方向であれば 2 を, 第 3 格子ベクトルの方向であれば 3 を指定します。11 行目, 12 行目は, Schrödinger 方程式を反復法で解くときに指定する必要がある数値で, 順に新しい解を混ぜる割合の初期値, 最終値, 反復回数, 収束判定誤差値にそれぞれ対応します。

## 11.3 計算の実際

### 11.3.1 入力ファイル F\_INP の作成について

ここでは, STM を用いて実際に STM 像のシミュレーションを行う際に必要な入力ファイル F\_INP 作成時における注意点を述べます。再度前項の入力ファイルを例に説明を行います。

#### PHASE 実行時に自動的に決定される項目

入力ファイル 1 行目の kv3, nspin, また次に続く  $k$  点の座標データ, 固有値の数 neg, 表面に垂直方向の結晶格子の長さ z1 は自動的に決まる値で, 変えることはできません。FFT の格子点の表面垂直方向の数に関係した値 nlpf は, PHASE 実行時に使った FFT の格子点の数の倍数であれば, 自由に設定できますが, 通常は同じ値に設定しておけば充分です。これらの量を調べるには PHASE を実行したディレクトリーにてコマンドプロンプトに

```
% grep STM output000
```

などと打てばよいです。上記のコマンドを実行した例を以下に記します。

```

!** tag_STM
!** sw_STM          =          1
!!STM:   kg(kng)     =    16411
!!STM:   kgp(kngp)   =    16411
!!STM:   kg1(kng1)   =    2088

```

```

!!STM:    neg(keg)  =      110
!!STM:    kimg      =        2
!!STM:    fft_box_size_WF(1, 1)(knl) =      90
!!STM:    fft_box_size_WF(2, 1)(knm) =      30
!!STM:    fft_box_size_WF(3, 1)(knn) =      16
!!STM:    fft_box_size_WF(1, 0)(kid) =      91
!!STM:    fft_box_size_CD(1, 1)(knlp) =      90
!!STM:    fft_box_size_CD(2, 1)(knmp) =      30
!!STM:    fft_box_size_CD(3, 1)(knp) =      16
!!STM:    fft_box_size_CD(1, 0)(kidp) =      91
!!STM:    kv3(knv3) =        4
!!STM:    nspin(kspin) =        1
!!STM:    == k-points ==
!!STM:    ik          CARTS                      PUCS
!!STM:    1  0.0000 -0.1624 -0.3247      0.0000 -0.3750 -0.3750
!!STM:    2  0.0000 -0.1624 -0.1082      0.0000 -0.3750 -0.1250
!!STM:    3  0.0000 -0.0541 -0.3247      0.0000 -0.1250 -0.3750
!!STM:    4  0.0000 -0.0541 -0.1082      0.0000 -0.1250 -0.1250
F_CNTN_STM = ./continue_bin_stm.data                                opened = false

```

### 指定が必要な項目

STM を使用するに当たってユーザー自身が指定する必要のあるパラメータは、エネルギーの範囲 ( $e1$ ,  $e2$ ), 真空領域の Schrödinger 方程式を解く範囲 ( $izi$ ,  $izf$ ), そしてソルバーの数値 ( $rini$ ,  $rfin$ ,  $nfin$ ,  $erlmt$ ) です。各項目の選び方について以下簡単に解説します。

- $e1$ ,  $e2$ : エネルギー範囲は Fermi エネルギー  $E_F$  より測って調べたいエネルギー範囲  $E_F + e1 < E < E_F + e2$  を指定すれば良いです。  $e1$ ,  $e2 < 0$  なら filled state 像,  $e1$ ,  $e2 > 0$  なら empty state 像が得られます。ただし、特にエネルギーの範囲に非占有状態 (empty state) が含まれる場合反復法の収束が悪くなり、場合によっては未収束となることがあります。
- $izi$ ,  $izf$ : この二つのパラメータは特に注意して選ぶ必要があります。まず、 $izi$  は表面からどれくらい離れたところで計算を始めるかを指定する数値です。表面より内側、あるいは表面ごく近傍に選ぶと、未収束となるか表面付近の電子状態が PHASE によるものと大きく変わってしまうかのいずれかの結果が得られるので注意が必要です。他方、 $izi$  が大きすぎると (表面から離れている場合大きな誤差を含む) PHASE による電子状態密度と変わらない結果となってしまいます。最適な  $izi$  はケース・バイ・ケースなので、ある程度の試行錯誤は必要ですが、おおよそ表面より  $\sim 1 \text{ \AA}$  程度離れた点に相当する  $izi$  を採用すると良いです。また、上記したエネルギー範囲が非占有状態を含む場合はさらに離れた (おおよそ 1.5 ~ 2 倍程度) 点に相当する  $izi$  を選ばないと計算が収束しません。他方、Schrödinger 方程式を解く領域の終点に相当する  $izf$  に関しては  $izi$  ほどの注意は要しませんが、反転対称がない場合を扱う際には若干の注意が必要です。すなわちこの場合、 $izf$  として系の底面に相当する高さの値を選ぶことが可能となりますので、それによる影響はない程度には小さな  $izf$  を採用する必要があると考えられます。
- $rini$ ,  $rfin$ ,  $nfin$ ,  $erlmt$ : これらの値は適切に選ぶことにより収束状況を変化させることができます。多くの場合新しい解を混ぜ合わせる割合は大きいほど収束は速くなるが、大きくしすぎると未収束になってしまいます。実際にはやはりある程度の試行錯誤が必要ですが、例えば上記した 0.1, 0.2, 100, 0.01 という組み合わせは収束にいたることが多いです。

#### 11.3.2 実行方法

まず、実行ファイル `stm` をコンパイルする必要があります。コンパイルの方法は PHASE などと同様です。すなわち、ソースディレクトリーへ移動した後、アーキテクチャーに応じたコマンドをタイプすればよいです。例えば Linux 上で Intel Fortran Compiler (ver.9.0) を使用するならば、

```
% make ifort
```



などとすれば良いです。しばらくしたら実行ファイル `stm` が生成されるはずです。

計算を行う際に必要な手順として、まず PHASE の入力ファイルの Postprocessing ブロック中に、以下のようなタグを挿入します。

```
STM{
    sw_stm = ON
}
```

このタグを挿入した状態で PHASE を実行することにより、STM が必要とするファイル群 (`continue_bin_stm.data`, `vlc.data` など、詳しくは 11.2.2 節) が生成されます。ついで、11.2.2 節にて説明している `stm_file_names.data` というファイルを編集し入出力ファイル名を指定し、11.3.1 節で説明したような STM 用入力ファイルを作成します。以上で準備が完了しますので、PHASE を実行したディレクトリー上で、実行ファイル `stm` を実行すればよいです。例えば PHASE を実行したディレクトリが `somedirectory/Si2x1`, 実行ファイル `stm` の場所が `somedirectory/bin/stm` だったとしたら、`somedirectory/Si2x1` に移動した後

```
% ../bin/stm
```

とすれば STM を実行することができます。

### 11.3.3 計算例

計算例として、Si ( $2 \times 1$ ) 系の計算を紹介します。図 16 に表面および表面から  $3.0 \text{ \AA}$  ほど離れた点での、STM と PHASE による電子状態密度を図示します。

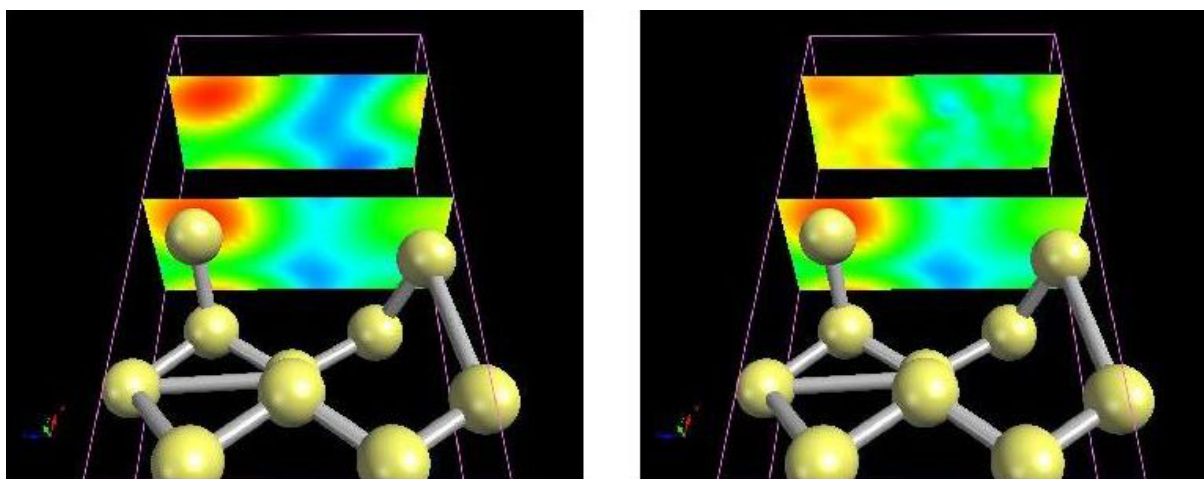


図 16: Si ( $2 \times 1$ ) の (100) 面の電子状態密度。表面および表面より  $3.0 \text{ \AA}$  ほど離れた点での結果を表示しています。左: STM によって精度良く計算した結果, 右: PHASE が出力した結果。色のスケールは見やすいよう適宜変えています。本図は、「革新的シミュレーションソフトウェアの研究開発」プロジェクトで公開している描画ソフト Biostation Viewer を用いて作成した。

この図から明らかなように、表面付近ではいずれの場合も電子状態密度を正しく評価できていますが、表面からある程度離れると PHASE による計算はノイズが大きくなるのに対し、STM による計算では正しい電子状態密度が得られています。このことより、表面からある程度離れた状況での電子状態密度の評価 (すなわち STM 像のシミュレーション) を行うためには、STM による処理が必要であると言えます。

## 12 電子状態解析：局所状態密度と部分電荷密度

### 12.1 はじめに

計算した電子状態を解析するため状態密度や電子密度を描きますが、複雑な物質になると解析が困難になります。原子領域の状態密度を求めることにより、結合状態の解析が可能となります。積層構造や界面構造の場合に層毎の状態密度を計算すると、積層による電子状態の変化の解析や界面状態の同定ができます。固有エネルギーがあるエネルギー範囲に収まる電子状態からなる部分電荷密度を計算すると、それらの電子状態の分布が明瞭に分かります。原子分割と層分割の局所状態密度と部分電荷密度の計算の仕方を BaO/Si(001) 界面を例に取り説明します。

簡単のため、BaO の格子定数に Si と同じ格子定数 (5.43Å) を用います。そして、図 17 に示すように、BaO/Si(001) 界面の原子構造は 5 層からなるシリコン層と 6 層からなる BaO 層を O で繋げた構造にします。この BaO/Si(001) 界面の計算の PHASE の入力 samples のディレクトリー BaO\_Si001 にあります。入力ファイル input\_BaO\_Si001.data の構造に関する部分は次のようになっています。

```
structure{
    unit_cell_type=bravais
    unit_cell{
        !! a_Si=5.43 A, c-axis=5*a_Si
        !! (c.f. a_BaO=5.52 A)
        !#units angstrom degree
        a = 3.83958982184, b= 3.83958982184, c= 27.15
        alpha=90.0, beta=90.0, gamma=90.0
    }

    symmetry{
        tspace{
            system = primitive
            generators {
                !#tag rotation tx ty tz
                E          0  0  0
                C2z        0  0  0
            }
        }
        sw_inversion = off
    }
    magnetic_state = para  !{para|af|ferro}

    atom_list{
        coordinate_system = internal ! {cartesian|internal}
        atoms{
            !#default mobile=no
            !#tag element rx ry rz num_layer
            Ba      0.0000 0.5000 0.05 1
            O       0.5000 0.0000 0.05 1
            Ba      0.5000 0.0000 0.15 2
            O       0.0000 0.5000 0.15 2
            Ba      0.0000 0.5000 0.25 3
            O       0.5000 0.0000 0.25 3
            O       0.0000 0.5000 0.35 4
            Si      0.0000 0.0000 0.40 5
            Si      0.5000 0.0000 0.45 6
            Si      0.5000 0.5000 0.50 7
            Si      0.0000 0.5000 0.55 8
            Si      0.0000 0.0000 0.60 9
            O       0.5000 0.0000 0.65 10
            Ba      0.5000 0.0000 0.75 11
            O       0.0000 0.5000 0.75 11
        }
    }
}
```

```

Ba      0.0000 0.5000 0.85 12
O       0.5000 0.0000 0.85 12
Ba      0.5000 0.0000 0.95 13
O       0.0000 0.5000 0.95 13
    }
}
element_list{ !#tag element  atomicnumber  zeta  dev
              Si              14  0.00  1.5
              Ba              56  0.00  1.5
              O               8  0.00  1.5
}
}

```

原子構造の緩和には時間がかかるので，mobile を no に設定して構造緩和は行わないことにします。

## 12.2 原子分割局所状態密度

原子分割の局所状態密度を計算するにはタグ Postprocessing の中にタグ dos とタグ ldos を書きます。そして，タグ dos の中の変数 sw\_dos を ON にし，タグ ldos の中の変数 sw\_aldos を ON にします。

```

Postprocessing{
  dos{
    sw_dos = ON
    method = g
  }
  ldos{
    sw_aldos = ON
    aldost{
      crtddst = 6.0 bohr
      naldost_from = 1
      naldost_to   = 19
    }
  }
}
}

```

タグ aldost の中の変数 crtddst は単位格子を原子ごとにボロノイ多面体分割するときの臨界距離です。どの原子からもこの臨界距離以上離れている領域は真空領域とみなされます。真空領域の状態密度は ( 原子の個数+1 ) 番目の原子局所状態密度として表されます。naldost\_from と naldost\_to に原子分割局所状態密度を計算する最初の原子と最後の原子を指定します。これを指定しないと全原子について原子分割局所状態密度が計算されます。また，タグ atoms の中で変数 aldost を off にした原子の局所状態密度は計算されません。変数 aldost よりも naldost\_from と naldost\_to の方が優先されます。

計算結果は dos.data に出力されます。状態密度図を作成するには，付属の Perl スクリプト dos.pl を使います。以下のようにすれば，dos\_a001.eps,dos\_a002.eps,...,dos\_axxx.eps といったポストスクリプトファイルが作成されます。

```
% ../../tools/bin/dos.pl dos.data -erange=-30,5 -dosrange=0,12 -mode=atom
```

dos.pl のオプションの詳細については PHASE TOOLS マニュアルを参照してください。

BaO/Si(001) 界面の原子分割局所状態密度を計算した結果を図 17 に示します。Si,Ba,O の原子分割局所状態密度にそれぞれの原子の特徴を見ることができます。

## 12.3 層分割局所状態密度

層分割の局所状態密度を計算するにはタグ Postprocessing の中にタグ dos とタグ ldos を書きます。そして，タグ dos の中の変数 sw\_dos を ON にし，タグ ldos の中の変数 sw\_layerdos を ON にします。

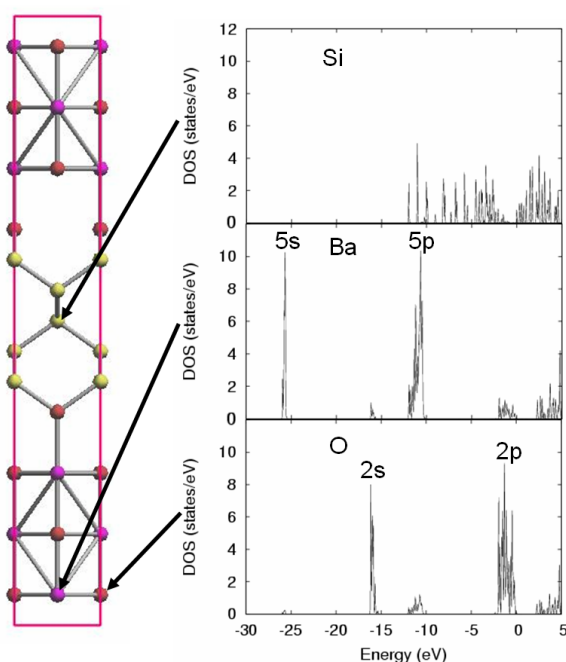


図 17: BaO/Si(001) 界面構造の原子分割の局所状態密度。上のパネル：Si 層中央の Si の局所状態密度。中央のパネル：BaO 層中央の Ba の局所状態密度。下のパネル：BaO 層中央の O の局所状態密度。

```
dos{
  sw_dos = ON
  method = g
}
ldos{
  sw_layerdos = ON
  layerdos{
    slicing_way = by_atomic_positions ![regular_intervals|by_atomic_positions]
    deltaz = 1.0 angstrom
    normal_axis = 3crtdst
    crtddst = 3.5 bohr
  }
}
```

タグ `layerdos` の中の変数 `normal_axis` では層分割するときの層の法線方向を指定します。1 が a 軸で、2 が b 軸で、3 が c 軸を表します。変数 `slicing_way` に `by_atomic_positions` を指定すると、原子位置によって局所状態密度を計算する層を定めることができます。この場合、タグ `atoms` で `num_layer` によって、原子が含まれる層の番号を指定します。先に示した、構造の入力部分では 13 個の層に各原子を割り当てています。変数 `slicing_way` に `regular_intervals` を指定すると、ある領域を等間隔に分割して作成した各層について局所状態密度を計算します。その間隔は変数 `deltaz` に入力します。変数 `crtddst` は層を作成する領域を決める臨界距離です。端の原子からこの臨界距離まで層を作成します。

層の範囲に関する下記のような記述が `output000` にあります。

!!ldos	no,	min,	max
!!ldos	1	0.00000000	5.13060607
!!ldos	2	5.13060607	10.26121214
!!ldos	3	10.26121214	15.39181821
!!ldos	4	15.39181821	19.23977276
!!ldos	5	19.23977276	21.80507579
!!ldos	6	21.80507579	24.37037883

```

!!ldos    7      24.37037883      26.93568186
!!ldos    8      26.93568186      29.50098489
!!ldos    9      29.50098489      32.06628793
!!ldos   10      32.06628793      35.91424248
!!ldos   11      35.91424248      41.04484855
!!ldos   12      41.04484855      46.17545462
!!ldos   13      46.17545462      51.30606069
!!ldos   14      0.00000000      0.00000000

```

no は層の番号です。min と max は層の下端の位置と上端の位置を示します。最後の層は指定した以外の領域です。

計算結果は dos.data に出力されます。状態密度図を作成するには、付属の Perl スクリプト dos.pl を使います。以下のようにすれば、dos\_l001.eps, dos\_l002.eps, ..., dos\_lxxx.eps といったポストスクリプトファイルが作成されます。

```
% ../../tools/bin/dos.pl dos.data -erange=-20,5 -dosrange=0,20 -mode=layer
```

dos.pl のオプションの詳細については PHASE TOOLS マニュアルを参照してください。

BaO/Si(001) 界面の層分割局所状態密度を計算した結果を図 18 に示します。

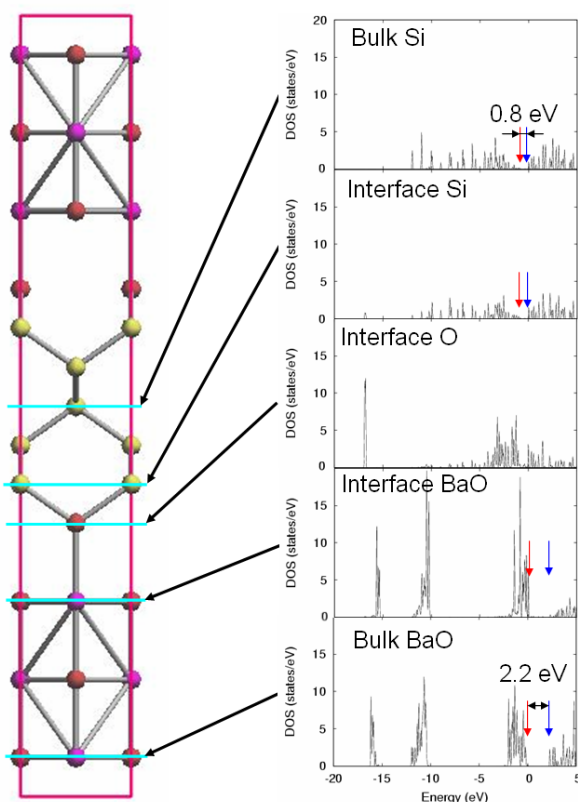


図 18: BaO/Si(001) 界面構造の層分割局所状態密度。一番上のパネル：Si 層の中央領域の局所状態密度。上から二番目のパネル：BaO/Si(001) 界面の Si 側の局所状態密度。中央のパネル：BaO/Si(001) 界面の酸素あたりの局所状態密度。下から二番目のパネル：BaO/Si(001) 界面の BaO 側の局所状態密度。一番下のパネル：BaO 層の中央領域の局所状態密度。

## 12.4 部分電荷密度

部分電荷密度を計算するにはタグ Postprocessing の中にタグ charge を書き、その中にタグ partial\_charge を書きます。そして、タグ charge の中の変数 sw\_charge\_rspace とタグ partial\_charge の中の変数 sw\_partial\_charge を On にします。変数 Erange\_min と Erange\_max にエネルギー領域の最大値と最小値を入力します。エネルギーは

金属の場合フェルミレベルから測り、絶縁体の場合は価電子帯上端のエネルギーから測ります。変数 `Erange_delta` に入力した値の間隔のエネルギー窓を先のエネルギー領域に作成します。このとき、先のエネルギー領域に適合するように、変数 `Erange_delta` の値は変更されます。出力ファイル `output000` には以下の様にエネルギー窓に関する出力があります。

```
!pc nEwindows = 20, nvb_windows = 10, ncb_windows = 10 <<m_ESoc_set_nEwindows_pc>>
!pc iw if_elec_state erange(hartree) erange(eV)
!pc (asis) (shifted) (shifted)
!pc 1 1 ( 0.094537 0.096374 ) ( -0.018375 -0.016537 ) ( -0.500000 -0.450000 )
!pc 2 1 ( 0.096374 0.098211 ) ( -0.016537 -0.014700 ) ( -0.450000 -0.400000 )
!pc 3 1 ( 0.098211 0.100049 ) ( -0.014700 -0.012862 ) ( -0.400000 -0.350000 )
!pc 4 1 ( 0.100049 0.101886 ) ( -0.012862 -0.011025 ) ( -0.350000 -0.300000 )
!pc 5 0 ( 0.101886 0.103724 ) ( -0.011025 -0.009187 ) ( -0.300000 -0.250000 )
!pc 6 1 ( 0.103724 0.105561 ) ( -0.009187 -0.007350 ) ( -0.250000 -0.200000 )
!pc 7 1 ( 0.105561 0.107399 ) ( -0.007350 -0.005512 ) ( -0.200000 -0.150000 )
!pc 8 0 ( 0.107399 0.109236 ) ( -0.005512 -0.003675 ) ( -0.150000 -0.100000 )
!pc 9 0 ( 0.109236 0.111074 ) ( -0.003675 -0.001837 ) ( -0.100000 -0.050000 )
!pc 10 1 ( 0.111074 0.112911 ) ( -0.001837 0.000000 ) ( -0.050000 0.000000 )
!pc 11 1 ( 0.112911 0.114749 ) ( 0.000000 0.001837 ) ( 0.000000 0.050000 )
!pc 12 0 ( 0.114749 0.116586 ) ( 0.001837 0.003675 ) ( 0.050000 0.100000 )
!pc 13 0 ( 0.116586 0.118424 ) ( 0.003675 0.005512 ) ( 0.100000 0.150000 )
!pc 14 0 ( 0.118424 0.120261 ) ( 0.005512 0.007350 ) ( 0.150000 0.200000 )
!pc 15 0 ( 0.120261 0.122099 ) ( 0.007350 0.009187 ) ( 0.200000 0.250000 )
!pc 16 1 ( 0.122099 0.123936 ) ( 0.009187 0.011025 ) ( 0.250000 0.300000 )
!pc 17 1 ( 0.123936 0.125773 ) ( 0.011025 0.012862 ) ( 0.300000 0.350000 )
!pc 18 0 ( 0.125773 0.127611 ) ( 0.012862 0.014700 ) ( 0.350000 0.400000 )
!pc 19 0 ( 0.127611 0.129448 ) ( 0.014700 0.016537 ) ( 0.400000 0.450000 )
!pc 20 0 ( 0.129448 0.131286 ) ( 0.016537 0.018375 ) ( 0.450000 0.500000 )
```

`nEwindows` はエネルギー窓の総数です。`nvb_windows` と `ncb_windows` はそれぞれ価電子状態と伝導電子状態を含むエネルギー窓の数です。`iw` はエネルギー窓の番号です。`if_elec_state` はそのエネルギー窓に電子状態があるかどうかを示しています。この値が 0 の時は電子状態がなく、1 の時には電子状態が存在します。列 `asis` には原子単位でエネルギー窓の範囲が示されています。二つの列 `shifted` にはエネルギーの基準から測ったときのエネルギー窓の範囲が原子単位と eV 単位で示されています。

変数 `partial_charge.filetype` に `individual` または `separate` を指定すると、各エネルギー窓ごとに計算された電荷密度が番号付けされたファイルに出力されます。その際の名前の付け方は、スピン分極がない場合であれば、`F_CHR = nfchr.cube` に対して `nfchr.00xx.cube` (`xx` には上の表の `iw` の値が入る) というようになります。スピン分極がある場合には、`F_CHR = nfchr.cube` に対して、`nfchr.up.00xx.cube`, `nfchr.down.00xx.cube` の二種類のファイルが生成されます。上の表で `if_elec_state` が 0 になっているのは、その範囲に固有値がある状態がないことを示しています。その場合、`cube` ファイルは生成されません。

`integrated` を選択すると各電荷密度がひとつのファイルに追記され、各電荷密度データの先頭には `PARTIALCHARGE` が記述され、終わりには `END` が記述されます。

BaO/Si(001) 界面の部分電荷密度を計算した結果を図 12.4 に示します。

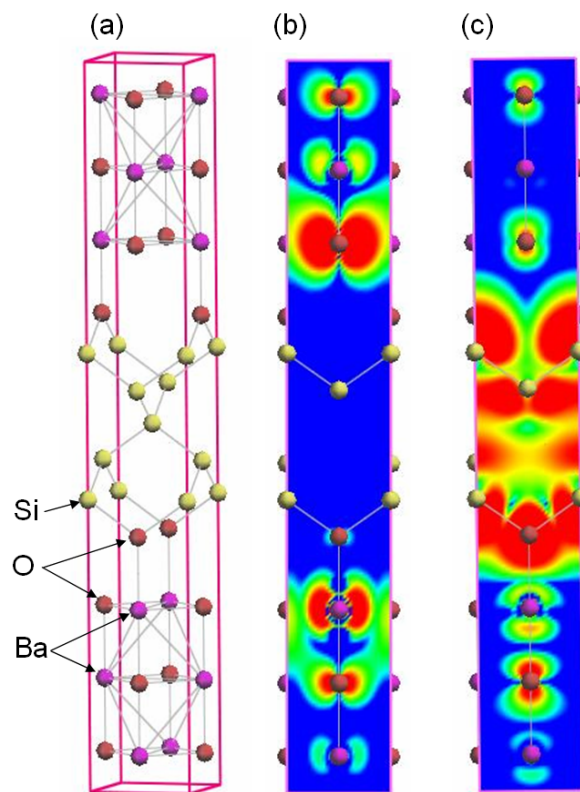


図 19: BaO/Si(001) 界面構造の部分電荷密度。(a) BaO/Si(001) 界面構造のモデル図。(b) フェルミレベル直下 (固有エネルギーが  $-0.05\text{eV}$  から  $0.0\text{eV}$  まで) の電子状態の部分電荷密度。(c) フェルミレベル直上 (固有エネルギーが  $0.0\text{eV}$  から  $0.05\text{eV}$  まで) の電子状態の部分電荷密度。電子密度は  $1 \times 10^{-5}$  から  $1 \times 10^{-3}$  までが示されています。青い部分には電子が少なく、赤い分には電子がたくさんあります。

## 13 DFT+U 法

### 13.1 はじめに

PHASE は密度汎関数理論に基づきほとんどの物質の電子状態は高精度に計算することができますが、密度汎関数法を適用する際に導入した局所密度近似は強相関電子系に対しては不十分で、不正確な電子状態を導きます。そのため、強相関電子系のオンサイトクーロン相互作用を補正の形で導入することが、しばしば行われます。このような補正を密度汎関数に加えた第一原理計算法を LDA+U 法または DFT+U 法と呼びます。本説明書では PHASE に実装されている DFT+U 法の理論と使用方法の説明します。

### 13.2 理論

DFT+U 法にはいくつかの方法がありますが、単純化された回転不変モデルを採用しました。DFT+U のエネルギー汎関数 ( $E_{\text{DFT+U}}$ ) には局所密度近似のエネルギー汎関数 ( $E_{\text{DFT}}$ ) に占有行列  $\rho$  の関数であるハバード補正項が加わります。

$$E_{\text{DFT+U}} = E_{\text{DFT}} + \frac{U_{\text{eff}}}{2} \sum_{I,m,\sigma} \left\{ \rho_{m,m}^{I\sigma} - \sum_{m'} \rho_{m,m'}^{I\sigma} \rho_{m',m}^{I\sigma} \right\}$$

占有行列はハバード補正を行う各サイトの原子軌道ごとに構成します。サイト I の占有行列はサイト I の局在軌道 (原子軌道や球面調和関数) に電子波動関数を射影することにより計算します。 $k$  は波数ベクトルで、 $n$  はバンド指標で、 $\sigma$  はスピン指標です。

$$\rho_{m,m'}^{I\sigma} = \sum_{k,n} f_{kn}^{\sigma} \langle \Psi_{kn}^{\sigma} | \phi_m^I \rangle \langle \phi_{m'}^I | \Psi_{kn}^{\sigma} \rangle$$

ここで、 $f_{kn}^{\sigma}$  は電子状態  $kn\sigma$  の占有数です。ハミルトニアンは DFT+U のエネルギー汎関数を密度行列で変分することにより求められます。有効クーロン相互作用  $U_{\text{eff}}$  の値は、実験に一致するように取るか、または、制限 DFT 計算から求めます。ハバード補正が行われた軌道は完全に占有している場合、その固有エネルギーより  $\frac{U_{\text{eff}}}{2}$  低い固有エネルギーをとり、完全に空いている場合は、その固有エネルギーより  $\frac{U_{\text{eff}}}{2}$  高い固有エネルギーをとります。

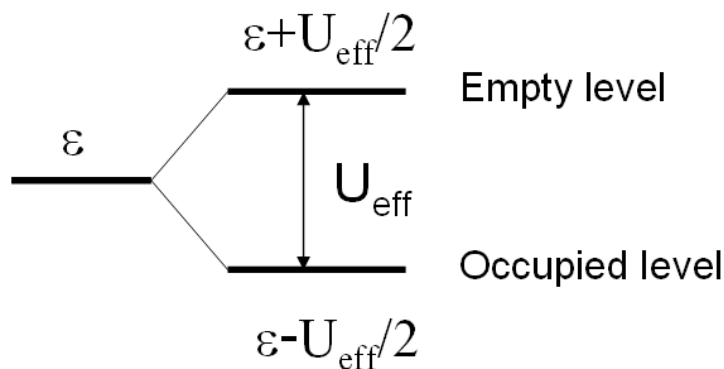


図 20: ハバード補正による軌道エネルギーの変化

### 13.3 使い方

DFT+U 法で電子状態を計算するには、通常の PHASE の入力に DFT+U 法の精度の指定とハバード補正を行うサイトの指定を書き加え、電子状態の解法の変更をする必要があります。まず、精度の指定は `accuracy` ブロック中に `hubbard` ブロックと `projector_list` ブロックを書き加えて、それらのブロック中の変数を設定することで行います。

```
accuracy{
```



```

...
hubbard{
  sw_hubbard = on
  projectors{
    #units eV
    #tag no ueff
      1 10.0
  }
}
projector_list{
  projectors{
    #tag no group radius l
      1 1 2.75 2
  }
}
...
}

```

ハバード補正を行うには `sw_hubbard` を `on` にしなければなりません。 `projectors` ブロック中にリストの形で、番号 `no` で特定されるプロジェクターに対して有効クーロン相互作用エネルギーの値 (`ueff`) を指定します。 `projector_list` 中の `projectors` ブロックにプロジェクターのリストを記述します。 `no` はプロジェクターの番号、`group` はプロジェクターのグループ番号、`radius` は有効原子半径、`l` は方位量子数です。

ハバード補正を行う原子を `atoms` のリストに変数 `proj_group` を書き加えて、使用するプロジェクターを指定します。

```

structure{
  ...
  atom_list{
    coordinate_system = internal ! {cartesian|internal}
    atoms{
      !#default mobile=no
      !#tag rx ry rz element proj_group
        0.0 0.0 0.0 Sr 0
        0.5 0.5 0.5 Ti 1
        0.0 0.5 0.5 0 0
        0.5 0.0 0.5 0 0
        0.5 0.5 0.0 0 0
    }
  }
  ...
}

```

`proj_group` で使用できる番号は `projector_list` で指定した `group` の値とハバード補正を行わないことを指定する `0` だけです。同種元素であれば、同じ `group` の値を設定します。異種元素に同じ `group` の値を設定することは出来ません。

DFT+U 法で電子状態を計算するときには、電子状態の解法を Davidson 法にかえることにより、正確に電子状態を求めることができます。 `solvers` ブロックでの指定は以下のようにすれば良いです。

```

wavefunction_solver{
  solvers{
    !#tag sol till_n dts dte itr var prec cmix
      Davidson -1 0.1 0.1 100 tanh off 1
  }
}

```

ここで説明したように入力を作成した後、`phase` を通常通りに実行します。出力には各回ごとに占有行列とそれを対角化したものが以下のように出力されます。

占有行列の添え字	$l = 0$	$l = 1$	$l = 2$	$l = 3$
1	$s$	$x$	$3z^2 - r^2$	$z(5z^2 - 3r^2)$
2		$y$	$x^2 - y^2$	$x(5z^2 - 3r^2)$
3		$z$	$xy$	$y(5z^2 - 3r^2)$
4			$yz$	$z(x^2 - y^2)$
5			$zx$	$xyz$
6				$x(x^2 - 3y^2)$
7				$y(3x^2 - y^2)$

表 9: 占有行列の添え字と軌道の性格の対応表

```

Occupation Matrix: is,ia,l=      1      2      2
 0.583   0.000   0.000   0.000   0.000
 0.000   0.583   0.000   0.000   0.000
 0.000   0.000   0.529   0.000   0.000
 0.000   0.000   0.000   0.529   0.000
 0.000   0.000   0.000   0.000   0.529
Diagonalizing Occupation Matrix: is,ia,l=      1      2      2
 0.529:   0.000   0.000   0.000  -1.000   0.000
 0.529:   0.000   0.000   1.000   0.000   0.000
 0.529:   0.000   0.000   0.000   0.000   1.000
 0.583:   0.000   1.000   0.000   0.000   0.000
 0.583:  -1.000   0.000   0.000   0.000   0.000

```

is はスピンの番号, ia は原子の番号, l は方位量子数です。行列の固有値は”:"の前に示され, 対応する行列の固有ベクトルはそのあとに示されます。

全エネルギーの出力とその成分の出力のあとにハバードエネルギー (HE) とハバードポテンシャルエネルギー (HP) が追加で出力されます。

```

TOTAL ENERGY FOR      2 -TH ITER=    -79.756461901287    edel =    0.482992D+01
KI=      45.2522902 HA=      125.6089055 XC=      -43.2979227 LO=      -147.0597534
NL=      19.3280980 EW=      -92.0686823 PC=      12.2272681 EN=      0.0000000
HE=      0.2533348 HP=      0.6709743

```

計算終了後の占有行列は occmat.data に出力されます。以下でこのファイルのデータフォーマットを説明します。一行目は占有行列の数  $N_{\text{om}}$  を記述します。

$$N_{\text{om}}$$

以下で説明する各占有行列の記述は, この数だけ繰り返します。占有行列の記述の初行では, スピン  $\sigma$ , 原子の番号  $i$ , 各原子ごとのプロジェクターの番号  $p$ , 原子種の番号  $\tau$ , 方位量子数  $l$  を順番に記述します。

$$\sigma \quad i \quad p \quad \tau \quad l$$

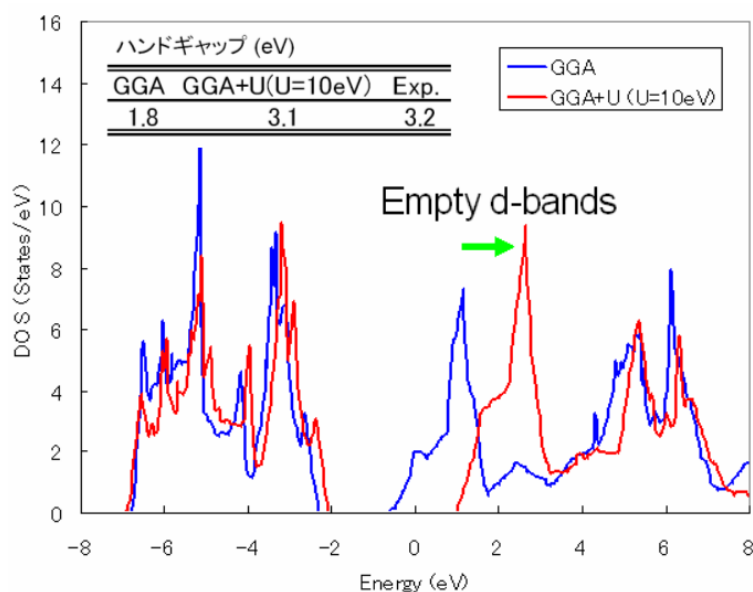
占有行列  $n^{\sigma,i,p}$  は  $2l + 1$  行  $2l + 1$  列の行列形式で記述します。

$$\begin{array}{cccc}
 n_{1,1}^{\sigma,i,p} & n_{1,2}^{\sigma,i,p} & \cdots & n_{1,2l+1}^{\sigma,i,p} \\
 n_{2,1}^{\sigma,i,p} & n_{2,2}^{\sigma,i,p} & \cdots & n_{2,2l+1}^{\sigma,i,p} \\
 \vdots & \vdots & \ddots & \vdots \\
 n_{2l+1,1}^{\sigma,i,p} & n_{2l+1,2}^{\sigma,i,p} & \cdots & n_{2l+1,2l+1}^{\sigma,i,p}
 \end{array} \quad (43)$$

行列の添え字は磁気量子数を表します。各方位量子数  $l$  での添え字と軌道の性格との対応を表 9 に示します。occmat.data は計算を継続数場合や, 占有行列の初期値を与える場合必要となります。

立方晶  $\text{SrTiO}_3$  の入力サンプルがサンプルディレクトリの下で DFT+U/SrTiO3/cubic+u ディレクトリにあります。この入力では, Ti の 3d 軌道間の有効クーロン相互作用  $U_{\text{eff}}$  を 10eV に設定してあります。DFT+U/SrTiO3/cubic ディレクトリには  $U_{\text{eff}}$  をゼロにした入力があります。これらの計算結果を図 21 に示します。

立方晶  $\text{LaVO}_3$  の入力サンプルが DFT+U/LaVO3/cubic+u ディレクトリにあります。この入力では, La の f 軌道間の  $U_{\text{eff}}$  を 20eV に設定してあります。DFT+U/LaVO3/cubic ディレクトリには  $U_{\text{eff}}$  をゼロにした入力があ

図 21: SrTiO<sub>3</sub> の状態密度

ります。 $U_{\text{eff}}$  がゼロのときは、f バンドはフェルミレベルの上 1.5eV 付近にあります。が、 $U_{\text{eff}}$  を 20eV とすることにより、8eV 付近に移ります。

より本格的な入力例として直方晶 LaVO<sub>3</sub> の入力サンプルを DFT+U/LaVO3/orthrombic+u ディレクトリに用意しました。ただし、原子構造は正確に実験データに合わせたわけではありませんので、実験のスペクトルを正しく再現しません。もし、必要ならば、正確な実験の原子構造で計算してください。この入力サンプルでは、La の f 軌道間の  $U_{\text{eff}}$  を 20eV に設定し、V の d 軌道間の  $U_{\text{eff}}$  を 5eV に設定してあります。この計算では V がスピン分極し、反強磁性状態になります。上向きスピンをもつ V と下向きスピンをもつ V は別元素として扱われます。DFT+U/LaVO3/orthrombic ディレクトリには  $U_{\text{eff}}$  をゼロにした入力があります。

occmat.data を使用して占有行列の初期値を入力する例として、FeO の電子状態計算のサンプル入力をディレクトリ DFT+U/FeO/gga+u に用意しました。この入力では、上向きスピンを持つ Fe の占有行列をの対角項をすべて 1 にし、下向きスピンを持つ Fe の占有行列の  $3z^2 - r^2$  成分のみを 1 としています。この占有状態で  $U_{\text{eff}}$  を 5eV とし、ハバード補正を行うことにより、GGA 近似では非占有であった  $3z^2 - r^2$  の性格を有する d バンドが占有され、バンドギャップが形成されます。DFT+U/FeO/gga ディレクトリには  $U_{\text{eff}}$  をゼロにした入力があります。

## 14 最大局在ワニエ関数

### 14.1 はじめに

電子状態は結晶全体に広がったブロッホ波と呼ばれる波として表されます。ブロッホ波を  $k$  空間に関してフーリエ変換することにより得られる局在した関数をワニエ関数と呼びます。ワニエ関数の中心位置は電子分布の平均位置を表すため、その和から結晶の分極を容易に知ることができます。また、ワニエ関数の自乗は電子の分布を表すため、化学結合に関する知見が得られます。一般にワニエ関数は一意に定まりません。ワニエ関数の広がりが最小になるように変換することにより得られる、一意に定まる関数を最大局在ワニエ関数といいます。

### 14.2 計算方法

最大局在ワニエ関数はワニエ関数の広がりを表す汎関数（広がり汎関数）を最小にするようにブロッホ波をユニタリー変換して求めます。広がり汎関数はユニタリー変換行列の関数で、広がり汎関数をユニタリー変換行列に関して微分して得られる行列はワニエ関数の広がりを狭めるユニタリー変換の方向になっています。この方向にわずかにユニタリー変換していくことで、ワニエ関数の広がりを最小にすることができます。

### 14.3 計算例:Si の最大局在ワニエ関数

最大局在ワニエ関数を計算するには、ポスト処理で Wannier 関数を計算することを指定する以下のような入力を記述します。

```
Postprocessing{
  wannier{
    sw_wannier = ON
    eps_grad = 1.d-3
    dt = 1.d-4
    max_iteration = 1000
    filetype = cube
  }
}
```

汎関数の勾配の大きさが `eps_grad` 以下になったら計算は終了します。dt は最急降下法の仮想的な時間刻みです。繰り返しが `max iteration` を超えたら計算は停止します。BioStationViewer などでも可視化するために、ワニエ関数の出力ファイルの形式を `cube` にしています。ワニエ関数のファイルの拡張子が `cube` になるように、以下のように入力に `file_names.data` にファイルポインタ `F_WANNIER` を記述します。

```
&fnames
F_INP = './nfinput.data'
F_POT(1) = './Si_ldapw91_nc_01.pp'
F_WANNIER = './nfwannier.cube'
&end
```

この機能は  $\Gamma$  点のみの計算でしか使用できません。

```
ksampling{
  method = gamma
}
```

また、並列計算には対応していません。並列計算で収束した結果を得ている場合は、非並列の継続処理としてワニエ関数の計算をしてください。Si の計算サンプルが `samples/wannier/Si8` にありますので、実行してみてください。計算が完了すると、`nfwannier.00001.cube` といったファイルが 16 個生成されます。それらの一つを BioStationViewer で可視化した結果が図 22 (a) に示されています。最大ワニエ関数は Si-Si 結合間に局在していることがわかります。これは、Si 結晶の結合様式が共有結合であることを示しています。

期待しない局所極小に収束してしまうことがありますが、`sw_random_wannier` を ON にしてランダムな状態から計算を始めることでこの問題が解決することがあります。

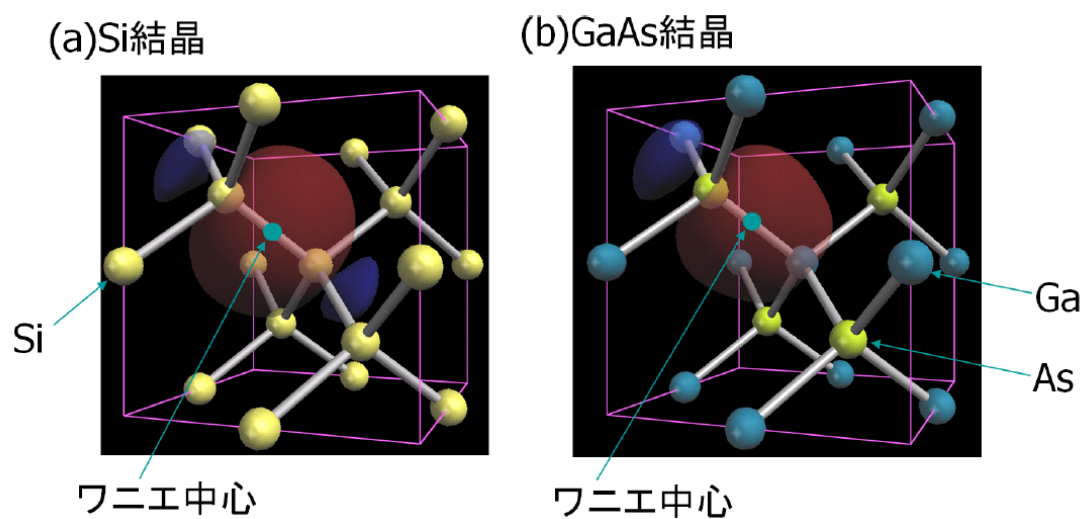


図 22: (a) Si 結晶と (b)GaAs 結晶の最大局在ワニエ関数

```
Postprocessing{
  wannier{
    ...
    sw_randomize = ON
    ...
  }
}
```

また，収束が不十分な状態で計算が終了しているようでしたら，`sw_continue` を ON にして計算を継続してください。

```
Postprocessing{
  wannier{
    ...
    sw_continue = ON
    ...
  }
}
```

## 15 陽電子寿命計算

### 15.1 はじめに

電子の反粒子である陽電子は、電子と同じ質量を持ち、正の電荷を持ちます。陽電子は電子と対消滅し、 $\gamma$  線を放出します。この陽電子消滅の現象を利用して、結晶の品質や、欠陥の研究が可能です。ここで、陽電子消滅実験から有用な情報を引き出すのに、第一原理計算に基づいて陽電子寿命を予測し、実験結果と比較することが重要です。PHASE には、完全結晶における陽電子寿命を予測する機能が付加されています。以下では、PHASE で採用されている計算手法の概要をのべ、つぎに具体的な計算のやり方について説明します。最後に、信頼できる計算を行うための注意点について述べます。

### 15.2 計算手法

PHASE では、次の手順で計算を行います。

- (A) はじめに通常の電子状態計算（バンド計算）を実行します。PHASE には、擬ポテンシャル・平面波法が実装されているので、この手法に基づいて計算を行います。バンド計算により、価電子の電荷密度  $\rho_v$  が得られます。全電子の電荷密度は次式で与えられます。

$$\rho_e = \rho_v + \rho_c \quad (44)$$

ここで、 $\rho_c$  は、コア電子の電荷密度です。CIAO で作成し公開されている擬ポテンシャルのデータファイルには、自由な原子におけるコア電子の電荷密度の情報が含まれています。このデータを読み込み (44) 式を評価します。

- (B) 陽電子波動関数  $\psi_+$  は次式で与えられます（原子単位）。

$$\left[ -\frac{1}{2}\Delta - \int dr' \frac{\rho_e(r') - \rho_n(r')}{|r - r'|} + \mu_c(\rho(r)) \right] \psi_+(r) = \varepsilon \psi_+(r) \quad (45)$$

ここで、 $\mu_c$  は、電子・陽電子相関に由来するポテンシャルエネルギーであり、 $\rho_n$  は、原子核の点電荷を表します。いま、陽電子は固体中に 1 個しかないと仮定するので、最も安定な固有状態のみを求めればよいことになります。したがって、陽電子の固有状態はブリルアンゾーン中の  $\Gamma$  点に属します。この波動関数を平面波によって展開します。

$$\psi_+ = \sum_{\vec{G}} C_{\vec{G}} \exp(i\vec{G} \cdot \vec{r}) \quad (46)$$

ここで、逆格子周期ベクトル  $\vec{G}$  の和を有限に抑えるため、平面波の運動エネルギーの上限を設定します。

- (C) (46) 式から、陽電子密度を求めます。

$$\rho_p(r) = |\psi_+|^2 \quad (47)$$

- (D) 電子及び陽電子の電荷密度を用いて、下記の式を評価し、陽電子寿命を計算します。

$$\frac{1}{\tau} = \pi r_e^2 c \int dr \rho_e(r) \rho_p(r) \Gamma(\rho_e) \quad (48)$$

ここで、 $r_e$  は、電子の古典半径、 $c$  は光速を表します。 $\Gamma$  は、増大因子であり、電子・陽電子間の相関に由来するものです。PHASE では、(48) 式の評価において、下記の近似を用いています。

$$\rho_e \Gamma(\rho_e) \cong \rho_v \Gamma(\rho_v) + \rho_c \Gamma(\rho_c) \quad (49)$$

この近似が成り立つためには、価電子とコア電子の分布の重なりが小さいことが条件となります。

以上、(A)～(D) において、計算の概略を述べました。本計算では、電子・陽電子間の相関に対して、局所密度近似を用いています。すなわち、相関ポテンシャルと増大因子は、均一な電子ガス中に陽電子が 1 個ある場合に計算された結果をもとに、電子密度の関数として与えられます。増大因子に関しては、次式が提案されています [5]。

$$\Gamma = 1 + 1.23r_s + 0.9889r_s^{3/2} - 1.482r_s^2 + 0.3956r_s^{5/2} + r_s^3/6 \quad (50)$$

ここで、 $\frac{4\pi}{3}r_s^3 = 1/\rho_e$  です。また、ギャップのある系、すなわち誘電体においては、金属よりも電子のスクリーニング効果が小さいため、次の補正を行う事を推奨します [6], [7]。

$$\Gamma = 1 + 1.23r_s + 0.9889r_s^{3/2} - 1.482r_s^2 + 0.3956r_s^{5/2} + (1 - 1/\varepsilon_{ele})r_s^3/6 \quad (51)$$

ここで、 $\varepsilon_{ele}$  は電子系誘電率です。その値が実験により測定されていない場合には、UVSOR により、密度汎関数理論に基づいて評価することができます。

### 15.3 具体的なやり方：Si 結晶の計算例

Si 結晶に対する計算例を示します。

#### 15.3.1 input file の解説

```
accuracy{
1.   cutoff_pwf = 50.00 rydberg
2.   positron_convergence{
3.     num_extra_bands = 8
4.     delta_eigenvalue = 1.d-8 rydberg
5.     succession = 6
6.     num_max_iteration = 32000
7.     dtim = 0.01
8.     epsilon_ele = 12
}
```

##### 1. cutoff\_pwf = 50.00 rydberg

これは、陽電子の波動関数を展開する際の [(46) 式参照] カットオフエネルギーです。

##### 2. positron\_convergence { }

このタグの中で、陽電子波動関数を反復計算によって求める、すなわち (45) 式を解く、際に、どのように収束解を得ようとするのか、その指定を行います。

##### 3. num\_extra\_bands = 8

陽電子の固有状態は、基底状態 1 個のみを計算すれば充分です。しかし、反復計算で収束解を得るには、それ以外に、基底状態よりもエネルギーの高い状態の波動関数も計算する必要があり、その個数を指定します。なお、求める波動関数は全てブリルアンゾーン中の  $\Gamma$  点に属します。

##### 4. delta\_eigenvalue = 1.d-8 rydberg

5. の説明参照

##### 5. succession = 6

反復計算において、前回と今回の物理量 (7. 参照) が 4. で与えられた範囲内で一致し、5. で指定された回数だけ連続して、この条件を満たせば、計算は収束したとみなします。

##### 6. num\_max\_iteration = 32000

計算は、この数繰り返すと、収束していなくても終了します。

##### 7. dtim = 0.01

繰り返し計算において、次の波動関数をどれだけ大きく変化させるかの尺度であり、dtim が大きいほど収束が早くなります。しかし、あまり大きいと収束解が得られなくなり、この値が小さいほど、安定に収束解が得られますが、小さくするほど収束が遅くなります。したがって、この値は計算する系により、ユーザーが最適な値を探すことを推奨します。

##### 8. epsilon\_ele = 12

epsilon\_ele は、ギャップのある系に対して、LDA の電子系誘電率補正を行う際に用いる tag です。= の後には、誘電率 (Si の場合 12) を用います。もし、誘電率の補正を行わないのであれば (たとえば、金属の計算を行う場合)、8. の行は削除します。

### 15.3.2 output file の解説

Si のサンプルプログラムにあるように、上記 input file を用いると、output000 ファイルと 3 個の cube file が生成されます。はじめに、output000 について解説します。このファイルの最初の部分は、Si の電子バンドの計算に関するものです。詳細については本マニュアルのほかの部分を参照してください。電子のバンド計算が終わり、電子の電荷密度が得られた後、陽電子の計算が行われます。出力では、次の部分からが陽電子計算に関するものです。繰り返し計算により、陽電子の波動関数が決定されます。下記の出力は、繰り返し計算のはじめにおいて、固有値が、14.6923a.u. であることを示しています。Extra bands はそれよりもエネルギーの高い固有値を表します。

```

--- initial positron energy eigen values ---
=== positron eigen values ===
  14.6923897769
-- extra_bands --
  15.0080438863    14.7287041392    15.0328744651    15.3199858426
  14.8514012423    14.6500235071    14.8527402877    15.0734017056

=== positron eigen values ===
  0.0498292501
-- extra_bands --
  0.1278903162    0.1473782963    0.2499012328    0.2578022704
  0.3535070925    0.3786636643    0.4035391484    0.5200959330

```

さらに、ファイルの下の方を見ていくと次の出力があります。

```

*****
positron lifetime(ps)  223.173562875375
core rate  3.63525329619323    %
*****

```

これは、最終的に陽電子の寿命が 223ps であることを示しています。Core rate は、全消滅速度に対するコア電子の消滅速度の割合を示します。

### 15.3.3 cube file の解説

計算が終了すると、electron.cube, positron.cube, ep\_pair.cube ファイルが出力されます。これらは、電子の電荷分布、陽電子の電荷分布、電子・陽電子ペアの分布を表す cube file であり、Biostation viewer (PHASE とは別にダウンロードして使う必要があります) を使って可視化できます。図 23 に Si 結晶における計算例を示します。価電子は主として、結合領域に存在し、陽電子は、隙間領域に存在することが分かります。陽電子の波動関数が広がり運動エネルギーが低下した方がエネルギー的に有利であることから、一般に陽電子は隙間領域に存在する傾向があります。電子・陽電子対分布を図 23(c) に示します。この分布が高いところで、陽電子が大きな確率で消滅することになります。

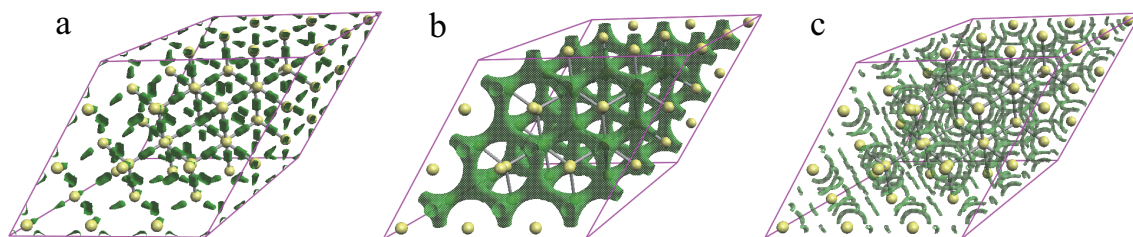


図 23: Si 結晶中の価電子分布 (a)、陽電子分布 (b)、ならびに、電子・陽電子対分布 (c)。

## 15.4 信頼性のある計算を行うために

陽電子寿命を計算するにあたり注意点をまとめます。



### 15.4.1 擬ポテンシャルの選択

元素によっては、セミコア状態を持つものがあります。ここで、セミコア状態とは、コア電子の内、軌道が空間的に広がり、コア電子の分布と価電子の分布との重なりが無視できない場合のことです。この場合、セミコア電子を価電子として取り扱い、擬ポテンシャルを作成することが望まれます。公開されている擬ポテンシャルには、元素によっては、このようにして作成されたものがあり、その際はこれを利用することを推奨します。もし、そのような擬ポテンシャルが作成されていなければ、CIAO を用いて作成することもできます。

### 15.4.2 カットオフエネルギーの選択

Si 結晶のバンド計算では、インプットファイルに次のように記されています。

```
accuracy{
  cutoff_wf = 50.00 rydberg ! cke_wf
  cutoff_cd = 200.00 rydberg ! cke_cd
  cutoff_pwf = 50.00 rydberg
```

これは、電子波動関数、電荷密度、陽電子波動関数に対するカットオフエネルギーです。これらの値を変化させて、計算された寿命が十分収束していることを確かめる必要があります。

### 15.4.3 陽電子波動関数の収束の確認

陽電子の波動関数は、電子の波動関数と同様、繰り返し計算によって求められます。各繰り返しにおいて output000 ファイルには、次のような出力があります。

```
=== positron eigen values ===
-0.5289133480
-- extra_bands --
-0.0108541347    -0.0064717757    -0.0059293552    0.0034291180
 0.0324778819    0.0547100102    0.0702660755    0.3261465565
lifetime:      223.168834154918      223.174883957176
```

ここで、計算終了近くで、陽電子の固有値 (positron eigen value) が十分収束していることを確かめます。サンプルプログラムでの出力では、

```
-0.5289133480
-0.5289133526
```

などとなっており、十分収束していることが分かります。また、上記の出力で 223.168 223.174 といった数値がありますが、前者は、価電子だけが消滅すると仮定した場合の寿命であり、後者は、コア電子も消滅することを考慮した寿命です。これらも、繰り返し計算の最後の方で十分収束していることを確かめる必要があります。通常の電子のバンド計算が十分収束しており、かつ 15.4.1, 15.4.2, 15.4.3 の事項を確認できれば、計算は十分考慮されたものであると考えてよいでしょう。まず、はじめに、比較的単純な系で計算を行い、計算が実験値を再現できているかどうかを確かめ、計算に十分になれたあとで、御自分が興味を持っている系の計算を行うことをお勧めします。本計算機能が、様々な系における陽電子消滅実験の解析に役立てば幸いです。

## 16 NEB 法

### 16.1 はじめに

Nudged Elastic Band (NEB) 法 [3] および Climbing Image (CI) NEB 法 [4] は、反応経路における始状態と終状態の間の最小エネルギー経路と鞍点を求める方法です。

NEB 法および CI-NEB 法を用いた反応経路計算においては、始状態の原子配置 ( $\vec{R}_0$ ) および終状態の原子配置 ( $\vec{R}_N$ ) が既知であるとして、始状態と終状態の間の中間状態 ( $\vec{R}_i$ ,  $i = 2 \sim N-1$ ) の原子配置やエネルギーなどを、隣接する状態 (イメージ) 間がばねによって結ばれているという拘束条件の下で構造最適化計算を行います。ここで  $\vec{R}_i$  は、各イメージにおける  $3M$  次元 ( $M$  は原子数) の座標です。NEB 法および CI-NEB 法の中間状態における初期原子配置は始状態と終状態から任意に決定することが可能ですが、始状態と終状態をイメージで等分割し決める方法が多くの場合採用されます。

### 16.2 理論

#### 16.2.1 作用力

通常の NEB 法 [3] 通常の NEB 法においては、各イメージの作用力は次のように与えられます。

$$\vec{F}_i = \vec{F}_i^s|_{\parallel} - \nabla E(\vec{R}_i)|_{\perp}. \quad (52)$$

ここで、 $\vec{F}_i^s|_{\parallel}$  は隣接するイメージ間のばねによる作用力の接線方向の成分であり、以下のように求められます。

$$\vec{F}_i^s|_{\parallel} = k \left( \left| \vec{R}_{i+1} - \vec{R}_i \right| - \left| \vec{R}_i - \vec{R}_{i-1} \right| \right) \cdot \hat{\tau} \hat{\tau}. \quad (53)$$

ここで  $k$  はばね定数です。 $\hat{\tau}$  は接線方向の単位ベクトルであり、次のように計算します。

$$\hat{\tau} = \frac{\vec{R}_i - \vec{R}_{i-1}}{\left| \vec{R}_i - \vec{R}_{i-1} \right|} + \frac{\vec{R}_{i+1} - \vec{R}_i}{\left| \vec{R}_{i+1} - \vec{R}_i \right|}. \quad (54)$$

(52) 式の  $\nabla E(\vec{R}_i)|_{\perp}$  は、第一原理計算などによって得られる、原子に働く力の接線に垂直な成分であり、以下のように求められます。

$$\nabla E(\vec{R}_i)|_{\perp} = \nabla E(\vec{R}_i) - \nabla E(\vec{R}_i) \cdot \hat{\tau} \hat{\tau}. \quad (55)$$

CI-NEB 法 [4] CI-NEB 法は、通常の NEB 法に対して最もエネルギーの高いイメージにおける作用力の計算方法を改良した方法です。CI-NEB 法計算では、通常の NEB 法を用いて反応経路計算をある程度進めた後に最も高いエネルギーのイメージ ( $\vec{R}_{i,\max}$ ) を決定し、 $\vec{R}_{i,\max}$  に働く作用力を次のように計算します。

$$\vec{F}_{i,\max} = -\nabla E(\vec{R}_{i,\max}) + 2\nabla E(\vec{R}_{i,\max}) \quad (56)$$

$$= \vec{F}_{i,\max}|_{\perp} - \vec{F}_{i,\max}|_{\parallel} \quad (57)$$

#### 16.2.2 ばね定数の計算方法 [4]

最少遷移エネルギーを求める反応経路計算においては、鞍点付近の計算精度を高くすることが好ましいと考えられます。このことから、鞍点付近でイメージの密度を高くし、接線の傾きを高い精度で求める必要があります。特に、反応経路全体と比較してポテンシャル障壁の領域が極めて狭い場合には、ポテンシャル障壁近傍のイメージ密度を高くすることにより高精度の計算を効率よく行うことが可能となります。最少エネルギー経路において鞍点付近にイメージを密に分布させる方法として、鞍点付近のばね定数  $k$  を大きくする方法が考えられています。NEB 法および CI-NEB 法における作用力は、ばねによる作用力とエネルギー計算によって得られた作用力の線形結合で表わされるので、イメージ間のばね定数は異なる値を選択することが可能です。ばね定数の設定方法としては、以下のエネルギーの線形関数が提唱されています。

$$k = k_{\max} - \Delta \left( \frac{E_{\max} - E_i}{E_{\max} - E_{\text{ref}}} \right) \quad (E_i \geq E_{\text{ref}} \text{ の場合}), \quad (58)$$

$$k = k_{\max} - \Delta k \quad (E_i < E_{\text{ref}} \text{ の場合}). \quad (59)$$

ここで,  $k_{\max}$  はばね定数の最大値,  $\Delta k$  はばね定数の最大値と最小値の差です。  $E_i$  は  $i$  番目のばねで結ばれた 2 つのイメージのうち高いエネルギーのイメージのエネルギー,  $E_{\max}$  は全イメージ中最も高いエネルギー,  $E_{\text{ref}}$  は始状態と終状態のうち, 高い方のエネルギーです。この  $E_{\text{ref}}$  の値の設定によって, 反応経路における始状態付近と終状態付近のイメージ密度が等しくなります。

## 16.3 使い方

### 16.3.1 実行方法

NEB は「レプリカ並列」に対応しています。以下のように起動します。

```
% mpirun -n NP phase ne=NE nk=NK nr=NR
```

ここで, NP は MPI プロセスの数, NR は並列で計算するレプリカの数, NE, NK は PHASE と同様バンドおよび k 点並列の数です。ただし,  $NP = NR \times NE \times NK$  という関係が成立している必要があります。

### 16.3.2 入力データ例

入力データの例を以下に示します。

```
Control{
  condition = initial ! {initial|continuation|automatic}
  cpumax = 1 day ! {sec|min|hour|day}
  max_iteration = 10000000
  multiple_replica_mode = ON
  multiple_replica_max_iteration = 2000
}
accuracy{
  cutoff_wf = 10.00 rydberg
  cutoff_cd = 40.00 rydberg
  num_bands = 28
  ksampling{
    method = monk ! {mesh|file|directin|gamma}
    mesh{ nx = 2, ny = 2, nz = 1 }
  }
  smearing{
    method = parabolic ! {parabolic|tetrahedral}
    width = 0.001 hartree
  }
  xctype = ggapbe
  scf_convergence{
    delta_total_energy = 0.5e-7 hartree
    succession = 2 !default value = 3
  }
  initial_wavefunctions = matrix_diagon !{random_numbers|matrix_diagon}
  matrix_diagon{
    cutoff_wf = 3.00 hartree
  }
}
structure{
  unit_cell_type = primitive
  unit_cell{
    a_vector = 10.400 0.000 0.000
    b_vector = 0.000 10.400 0.000
    c_vector = 0.000 0.000 30.200
  }
  symmetry{
    sw_inversion = off
  }
  atom_list{
    coordinate_system = cartesian ! {cartesian|internal}
    atoms{
      #units angstrom
      #tag element rx ry rz mobile
      Si 0.000000000000 0.000000000000 0.000000000000 0
      Si 2.751721694800 2.751721694800 0.000000000000 0
      Si 1.375860847400 1.375860847400 1.375860847400 0
      Si 4.127582542200 4.127582542200 1.375860847400 0
      Si 0.000000000000 2.751721694800 2.751721694800 0
      Si 2.751721694800 0.000000000000 2.751721694800 0
    }
  }
}
```

```

Si    1.375860847400    4.127582542200    4.127582542200    0
Si    4.127582542200    1.375860847400    4.127582542200    0
Si    0.000000000000    0.000000000000    5.503443389600    0
Si    2.751721694800    2.751721694800    5.503443389600    0
Si    1.375860847400    1.375860847400    6.879304237000    0
H     1.644706293661    1.095414892118    11.000000000000    1
H     1.095414929519    1.644706317263    11.000000000000    1
}
}
element_list{
  #tag element atomicnumber mass zeta dev
  #units atomic_mass
  Si      14      28.085
  H        1       1.008
}
}
multiple_replica{
  method = nudged_elastic_band_method
  accuracy{
    dt = 40 au_time
    neb_time_integral = quench
    penalty_function = off
    neb_convergence_condition = 3
    neb_convergence_threshold = 5.0e-04
  }
  constraint{
    ci_neb = OFF
    sp_k_init = 0.03
    sp_k_min = 0.03
    sp_k_max = 0.03
    sp_k_variable = OFF
  }
  structure{
    number_of_replicas = 6
    replicas{
      #tag replica_number howtogive_coordinates end0 end1
      1      proportional      0      -1 ! 0: end0, -1:end1
      2      proportional      0      -1
      3      proportional      0      -1
      4      proportional      0      -1
      5      proportional      0      -1
      6      proportional      0      -1
    }
    endpoint_images = directin ! {no or nothing | file | directin}
    howtogive_coordinates = from_endpoint_images
    atom_list_end0{
      coordinate_system = cartesian ! {internal|cartesian}
      atoms{
        #units angstrom
        #tag element rx ry rz
        Si    0.000000000000    0.000000000000    0.000000000000
        Si    2.751721694800    2.751721694800    0.000000000000
        Si    1.375860847400    1.375860847400    1.375860847400
        Si    4.127582542200    4.127582542200    1.375860847400
        Si    0.000000000000    2.751721694800    2.751721694800
        Si    2.751721694800    0.000000000000    2.751721694800
        Si    1.375860847400    4.127582542200    4.127582542200
        Si    4.127582542200    1.375860847400    4.127582542200
        Si    0.000000000000    0.000000000000    5.503443389600
        Si    2.751721694800    2.751721694800    5.503443389600
        Si    1.375860847400    1.375860847400    6.879304237000
        H     1.644706293661    1.095414892118    11.000000000000
        H     1.095414929519    1.644706317263    11.000000000000
      }
    }
    atom_list_end1{
      coordinate_system = cartesian ! {internal|cartesian}
      atoms{
        #units angstrom
        #tag element rx ry rz
        Si    0.000000000000    0.000000000000    0.000000000000
        Si    2.751721694800    2.751721694800    0.000000000000
        Si    1.375860847400    1.375860847400    1.375860847400
        Si    4.127582542200    4.127582542200    1.375860847400
        Si    0.000000000000    2.751721694800    2.751721694800
        Si    2.751721694800    0.000000000000    2.751721694800
      }
    }
  }
}

```

```

      Si    1.375860847400      4.127582542200      4.127582542200
      Si    4.127582542200      1.375860847400      4.127582542200
      Si    0.000000000000      0.000000000000      5.503443389600
      Si    2.751721694800      2.751721694800      5.503443389600
      Si    1.375860847400      1.375860847400      6.879304237000
      H     2.22686927      0.48813212      7.65400988
      H     0.48813224      2.22686933      7.65400957
    }
  }
}
wavefunction_solver{
  solvers{
    #tag sol till_n dts dte itr var prec cmix submat
    lmMSD -1 0.2 0.2 1 linear on 1 on
  }
}
charge_mixing{
  mixing_methods{
    #tag no method rmxs rmxe itr var prec istr nbmix update
    1 broyden2 0.10 0.10 1 linear on 1 0 RENEW
  }
}
printoutlevel{
  base=1
}

```

### 16.3.3 NEB 関連ファイル

NEB 関連のファイルは, `file_names.data` で設定します。次のように記述します。

```

&fnames
F_INP='./nfinp.data'
F_POT(1)='./Si_ggapbe_nc_01.pp'
...
...
/
&nebfiles
F_IMAGE(0) = './endpoint0.data'
F_IMAGE(-1) = './endpoint1.data'
F_NEB_OUT = './output_neb'
F_NEB_ENF = './nfnebenf.data'
F_NEB_DYNM = './nfnebdynm.data'
/

```

ファイル読み込みの `namelist` として, `&nebfiles` を利用している点にご注意ください。

`&nebfiles` で利用できるファイルポインターを, 表 10 に示します。

### 16.3.4 入力データ (NEB 関連のキーワード)

入力データの識別子 NEB に関連する, 入力データの識別子およびその説明を表 11 に示します。

**収束判定** 収束判定条件は, 変数 `neb_convergence_condition` で設定します。`neb_convergence_condition` には, 数値または文字列を指定することができます。表 12 に, 設定できる条件を示します。

**ストップ, リスタート**

**計算のストップ** 入力データの `multi_replica_max_iteration`, または `nfnebstop.data` に記述された NEB の `iteration` 数で NEB 計算は終了します。また, 各イメージの電子状態計算において, 入力データの `max_iteration`,

表 10: NEB で利用できるファイルポインター

ファイル名変数	Unit 番号	デフォルト値	備考
F_IMAGE(-1:99)	201	./endpoint0.data (F_IMAGE(0)) ./endpoint1.data (F_IMAGE(1))	イメージの原子座標
F_NEB_STOP	202	./nfnebstop.data	NEB ステップ終了用 ファイル
F_NEB_OUT	203	./output_neb	NEB 計算 ログ出力
F_NEB_CNTN	204	./neb_continue.data	NEB 継続計算用 ファイル
F_NEB_ENF	205	./nfnebenf.data	エネルギー, 力 出力ファイル
F_NEB_DYNM	206	./nfnebdynm.data	原子座標 出力ファイル

表 11: NEB に関連する入力データ

第 1 ブロック 識別子	第 2, 第 3 ブロック 識別子	タグ識別子	説明
Control			
		multiple_replica_mode	NEB 計算の実行 ON,OFF
		multiple_replica_max_iteration	NEB iteration 数
multiple_replica			
		method	反応経路計算手法 nudged_elastic_band_method
	accuracy		
		dt	NEB 計算における原子座標更新の $\Delta t$
		neb_time_integral	時間積分法 quench, steepest_descent(デフォルト)
		penalty_function	ペナルティ関数 ON,OFF
		neb_convergence_condition	NEB 収束判定法 (後述)
		neb_convergence_threshold	NEB 収束判定値
	constraint		
		ci_neb	CI-NEB ON, OFF
		sp_k_init	ばね定数 (初期値)
		sp_k_min	ばね定数 (最小値)
		sp_k_max	ばね定数 (最大値)
		sp_k_variable	ばね定数の固定, 変動 OFF 固定, ON 変動
	structure		
		number_of_replicas	レプリカ数
	replica		レプリカ情報
		endpoint_images	両端のイメージの指定方法 directin, file
	atom_list_end0		両端のイメージの原子リスト
	atom_list_end1		両端のイメージの原子リスト

表 12: 収束判定条件設定のパラメーター

数値	文字列	説明
1	energy_e	dE < threshold
2	phase_force	PHASE の力の最大値 < threshold
3	neb_force	NEB 計算で補正した力の最大値 < threshold
4	force_at_transition_state	最大エネルギーイメージの PHASE の力の最大値 < threshold
5	phase_force_normal	PHASE の、経路に垂直な成分の力の最大値 < threshold

cpumax, nfstop.data ファイルの設定によっても計算は終了します。いずれの場合でも、停止した箇所からリスタートすることが可能です。

ここで、計算ストップ時における通常の PHASE との相違点を挙げます。PHASE では、nfstop.data ファイルによって終了した場合、nfstop.data ファイルは空ファイルとなります。他方 NEB 計算では、あるイメージを nfstop.data によって終了した場合、nfstop.data はただちには空ファイルとはならず、ほかのイメージの計算を行います。NEB 計算終了処理においてはじめて nfstop.data ファイルを空ファイルとします。

計算のリスタート PHASE と同様、入力データにおいて、condition の値を continuation とすることによってリスタート計算を行います。

```
Control{
  condition = continuation
  ...
  ...
}
```

リスタート時に利用するファイルは下記の通り。

- NEB 計算: neb\_continue.data
- 電子状態計算: 各レプリカの PHASE 用リスタートファイル;  
neb\_continue.data, continue.data\_r\*, continue\_bin.data\_r\*, zaj.data\_r\*, nfchgt.data\_r\*

両端のイメージの原子座標をファイルから指定する方法 両端のイメージの原子座標をファイルで指定する場合は、入力データの endpoint\_images の値を file とし、file\_names.data にイメージのファイル名を設定します。その際、file\_names.data ファイル中では F\_IMAGE(-1) および F\_IMAGE(0) というファイルポインターを利用します。以下に、入力データと file\_names.data の記述例を記します。

入力データの記述例

```
multiple_replica{
  ...
  ...
  structure{
    endpoint_images = file
  }
  ...
  ...
}
```

file\_names.data ファイルの記述例

```
&fnames
...
...
/
```

```
&nebfiles
F_IMAGE(0) = './endpoint0.data'
F_IMAGE(-1) = './endpoint1.data'
...
...
/
```

また，原子座標データファイル（上記の例では endpoint0.data や endpoint1.data というファイル名）は，次のような形式で記述します。

```
coordinate_system=cartesian
```

```
#units angstrom
```

```
Si      0.000000000000      0.000000000000      0.000000000000
Si      2.751721694800      2.751721694800      0.000000000000
Si      1.375860847400      1.375860847400      1.375860847400
Si      4.127582542200      4.127582542200      1.375860847400
Si      0.000000000000      2.751721694800      2.751721694800
Si      2.751721694800      0.000000000000      2.751721694800
Si      1.375860847400      4.127582542200      4.127582542200
Si      4.127582542200      1.375860847400      4.127582542200
Si      0.000000000000      0.000000000000      5.503443389600
Si      2.751721694800      2.751721694800      5.503443389600
Si      1.375860847400      1.375860847400      6.879304237000
H       1.644706293661      1.095414892118      11.000000000000
H       1.095414929519      1.644706317263      11.000000000000
```

中間イメージの原子座標 中間イメージの原子座標は，以下の 2 通りの方法で指定することができます。

1. 両端の原子座標から，線形補間で作成 (proportional)
2. 中間イメージの座標値を，ファイルで指定する (file)

中間イメージをファイルで指定する場合は，replica タグ内の howtogive\_coordinates を file とし，対応する原子座標ファイルは file\_names.data ファイルで指定します。入力データと file\_names.data ファイルの記述例を以下に示します。

入力データの記述例

```
multiple_replica{
...
...
  structure{
    number_of_replicas = 3
    replicas{
      #tag replica_number howtogive_coordinates end0 end1
      1          file          0      -1 ! 0: end0, -1:end1
      2          file          0      -1
      3          file          0      -1
    }
  }
}
```

file\_names.data の記述例

```
&fnames
...
```



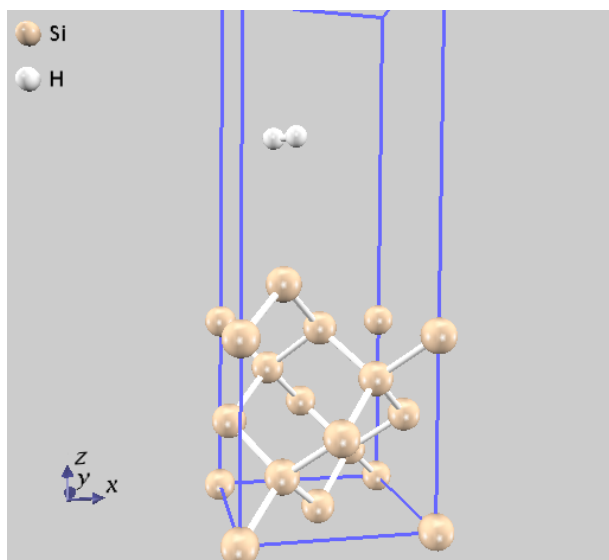


図 24: 本例題の始状態。

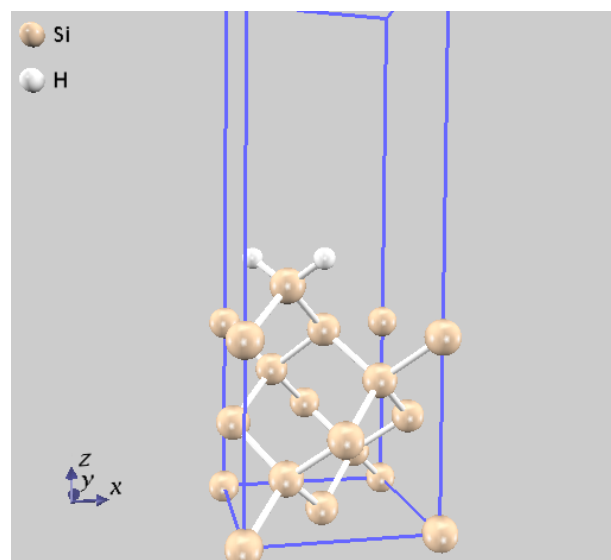


図 25: 本例題の終状態。

```
...
/
&nebfiles
F_IMAGE(0) = './endpoint0.data'
F_IMAGE(-1) = './endpoint1.data'
F_IMAGE(1) = './image1.data'
F_IMAGE(2) = './image2.data'
F_IMAGE(3) = './image3.data'

/
```

原子座標データを指定するファイルの書式は、両端イメージの場合と同じです。

## 16.4 例題

本機能を利用した例題について説明します。例題の入力ファイルは、`samples/neb` 以下にあります。

ここで紹介する例題は、シリコン表面に水素分子が解離吸着する反応をシミュレートします。始状態は表面と表面から十分離れた場所にある水素分子から成る系、始状態は表面のシリコン原子に水素分子が解離し、吸着した系です。始状態と終状態の構造をそれぞれ図 24 と図 25 に示します。ただし、あくまで例題ですので、通常は実行する始状態、終状態の構造最適化は実行していません。

### 16.4.1 入力ファイルの説明

入力ファイルについて簡単に説明します。

`control` ブロックにおいて、全体的な計算条件の指定を行います。

```
Control{
  condition = initial ! {initial|continuation|automatic}
  cpumax = 1 day ! {sec|min|hour|day}
  max_iteration = 10000000
  multiple_replica_mode = ON
  multiple_replica_max_iteration = 2000
}
```

multiple\_replica\_mode に ON を指定することにより, NEB の計算が実行されます。また, NEB の繰り返し計算の上限回数を multiple\_replica\_max\_iteration 変数によって 2000 としています。

multiple\_replica ブロックの下 structure ブロックにおいてレプリカの指定を実行しています。以下のようになります。

```
multiple_replica{
  ....
  structure{
    number_of_replicas = 6
    replicas{
      #tag replica_number  howtogive_coordinates  end0  end1
      1          proportional          0      -1 ! 0: end0, -1:end1
      2          proportional          0      -1
      3          proportional          0      -1
      4          proportional          0      -1
      5          proportional          0      -1
      6          proportional          0      -1
    }
    endpoint_images = directin ! {no or nothing | file | directin}
    howtogive_coordinates = from_endpoint_images
    atom_list_end0{
      coordinate_system = cartesian ! {internal|cartesian}
      atoms{
        #units angstrom
        #tag element rx ry rz
        Si  0.000000000000      0.000000000000      0.000000000000
        ...
        ...
      }
    }
    atom_list_end1{
      coordinate_system = cartesian ! {internal|cartesian}
      atoms{
        #units angstrom
        #tag element rx ry rz
        Si  0.000000000000      0.000000000000      0.000000000000
        ...
        ...
      }
    }
  }
  ....
}
```

number\_of\_replicas に 6 と指定していますが, この指定によってレプリカ数を合計 6 としています。replicas ブロックにおいて実際にどのようにレプリカの座標を作るかを指定しています。この例では, すべて始状態・終状態の線形補完によって作る, という設定になります。atom\_list\_end0 および atom\_list\_end1 ブロックには始状態・終状態の座標値を指定しています。この指定は, 前記の通り通常の PHASE の座標指定と変わるところはありません。

multiple\_replica ブロックの下 accuracy ブロックには, 以下のような指定を行っています。

```
multiple_replica{
  ...
  accuracy{
    dt = 40 au_time
    neb_time_integral = quench
    penalty_function = off
  }
}
```

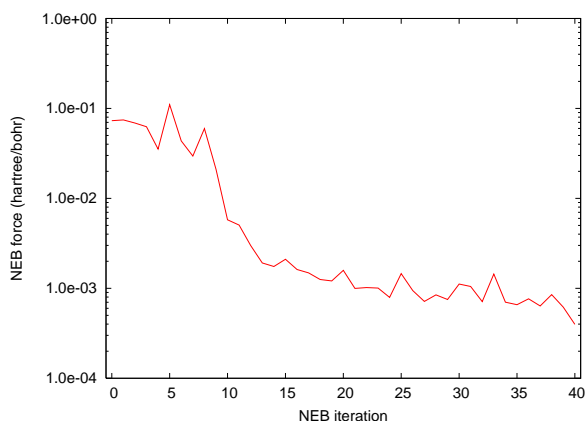


図 26: NEB 力の履歴。

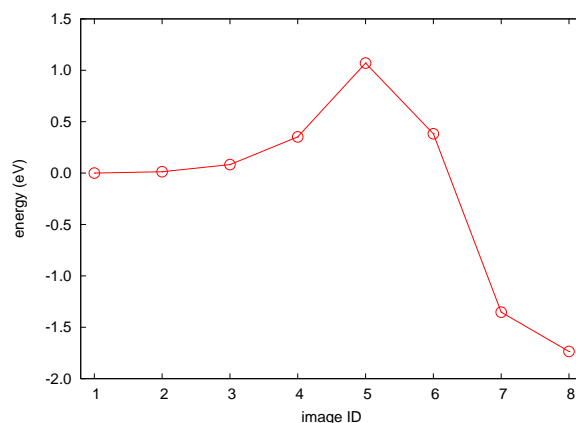


図 27: 最終的に得られる反応経路と各レプリカのエネルギーの関係。

```

    neb_convergence_condition = 3
    neb_convergence_threshold = 5.0e-04
}
}

```

このような指定を行うと、レプリカを時間発展させる方法として、quenched MD 法を採用します。時間刻みは原子単位で約 40(約 0.96 ps) となります。また、収束判定は NEB 力を利用し、 $5 \times 10^{-4}$  hartree/bohr 以下となったら収束したとみなします。

#### 16.4.2 計算結果

本例題を実行すると得られる結果を紹介します。

図 26 に、本例題を実行すると得られる、NEB の繰り返し計算と NEB 力の最大値の関係を示します。はじめのうちは大きな力が働いていますが、計算が進行するにつれて小さくなっていき、41 回の繰り返し計算の後収束判定を満たして計算が終了しています。

図 27 に、本例題を実行すると得られる各イメージとエネルギーの関係を示します。この図より、遷移状態は 4 番目のレプリカであり、始状態から見ると障壁エネルギーが約 1.08 eV であることが分かります。

図 28 に、遷移状態における原子配置を示します。この図から明らかなように、本例題では「水素分子が解離、そして吸着する」直前の構造が遷移状態です。

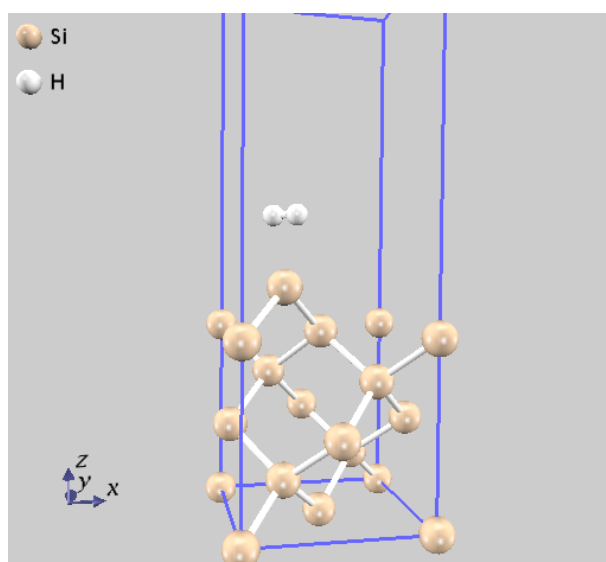


図 28: 遷移状態における原子配置。

## 17 拘束条件付きダイナミクスと Blue Moon 法

### 17.1 はじめに

化学反応経路を探索する手法として、ボンド長やボンド角などの化学反応を特徴づける「反応座標」を導入し、想定した反応経路上でその値を逐次変化させながら反応座標を拘束した構造最適化や分子動力学シミュレーションを実施する、という手法があります。単純な構造最適化の場合絶対零度における反応経路が得られ、有限温度の分子動力学シミュレーションを実施すると自由エネルギー差が得られます。ここでは、PHASE を利用して拘束条件付きダイナミクスを追跡する方法を説明します。

### 17.2 拘束条件付きダイナミクスの基本的な使い方

#### 17.2.1 入力ファイルの記述方法

拘束条件付きダイナミクスを実行するには、まず以下の要領で condition ブロックの下で driver 変数を指定します。

```
condition{
  ...
  driver=constraints
  ...
}
```

次に、以下のように structure ブロックの下に constrainablexx ブロックを作成します。ここで xx は整数です。

```
structure{
  ...
  ...
  constrainable1{
    type=bond_length
    atom1=1
    atom2=2
    mobile = off
    monitor = off
    reaction_coordinate{
      sw_reaction_coordinate=on
      init_value = 2.4 angstrom
      increment = 0.1 angstrom
      final_value = 8.0 angstrom
    }
    plane{
      normx=1
      normy=0
      normz=0
    }
    coordination_number{
      kappa = 5.0
      rc = 2.0 angstrom
    }
  }
  ...
  ...
}
```

拘束条件はいくつでも指定可能ですが、たとえば constrainable1, constrainable2, constrainable4 という 3 つの constrainablexx ブロックが存在する場合、constrainable4 ブロックは入力解釈の対象にはなりません。また、互いに相いれない拘束条件を定義してしまうと、拘束条件を課すための収束計算が破たんしてしまう場合がありますので注意が必要です。constrainablexx ブロックの下では、以下の変数/ブロックを指定することができます。

type 変数 拘束条件の“種類”を指定します。以下のいずれかの値をとります。

bond\_length 2 原子間の距離を拘束します。

bond\_angle 3 原子の成すボンド角を拘束します。

dihedral\_angle 4 原子の 2 面角を拘束します。

bond\_length\_diff 2 原子間の距離の差を拘束します。

bond\_angle\_diff 3 原子が成す角度の差を拘束します。

distance\_from\_pos 指定の場所からの距離を拘束します。

plane 面内に指定の原子を拘束します。

center\_of\_mass 指定の原子群の重心を拘束します。

coordination\_number 配位数を拘束します。ここで、0 番目の原子の周りの「配位数」は以下の式によって評価します。

$$\sigma = \sum_{i \neq 0} S(|\mathbf{r}_i - \mathbf{r}_0|),$$

$$S(r) = \frac{1}{\exp[\kappa(r - r_c)] + 1} \quad (60)$$

$\kappa$ ,  $r_c$  はパラメーターであり、first coordination shell でほどよく 0 に近づくように決定します。

atomx 変数 指定の拘束条件に関わる原子を指定します。x は数字であり、たとえば 2 原子間の距離の場合は 2 つの原子が拘束に関わるので、atom1 と atom2 に対応する原子の番号を指定します。type が coordination\_number の場合、配位数を計算する中心の原子の番号を指定します。

mobile 変数 指定の拘束条件が“可動か否か”を指定するためのスイッチです。off とすると拘束され、on とすると拘束はされません。デフォルト値は off です。

monitor 変数 指定の拘束条件を“監視”(値を算出し、ログファイルに出力)するか否かを指定するスイッチです。デフォルト値は off です。

reaction\_coordinate ブロック 指定の拘束条件が、“反応座標(逐次値を変化させられる)”である場合に作成するブロックです。以下の変数を指定することができます。

sw\_reaction\_coordinate on の際、反応座標とみなされます。

init\_value 反応経路の初期値を、対応する単位で指定します。指定がない場合、入力原子配置から求められる値が採用されます。この値と、入力原子配置から求められる値が異なる場合、入力原子配置が修正されたのちに計算が実行されます。このため、一回目の(拘束力も含む)原子に働く力の最大値が見掛け上非常に大きな値となることがありますが、これは正常な振る舞いです。

final\_value 反応経路の最終値を対応する単位で指定します。

increment final\_value と init\_value の間の刻み幅を指定します。

反応座標を逐次変化させる場合、以下のケースは特殊であるので注意が必要です。

type が plane の場合 この場合、変化するのは面の原点です。原点の座標値は、指定の法線ベクトルと原子の座標値からプログラムが自動的に決めますが、反応座標の変化としては、法線ベクトルの方向に原点が変化するという振る舞いになります。init\_value, final\_value, increment は、この原点の移動量を指定してください。なお、この場合の init\_value のデフォルト値は 0 ですが、通常明示的に指定する必要はないはずです。

type が center\_of\_mass の場合 この場合、重心を、指定の方向に移動させます。init\_value, final\_value, increment は、この移動量を指定してください。なお、この場合の init\_value のデフォルト値は 0 ですが、通常明示的に指定する必要はないはずです。

plane ブロック 面内拘束の場合の、拘束したい面の法線ベクトルを指定するためのブロックです。次の変数を指定することができます。

normx 法線ベクトルの x 座標。

**normy** 法線ベクトルの  $y$  座標。

**normz** 法線ベクトルの  $z$  座標。

**distance\_from\_pos** ブロック type として **distance\_from\_pos** を採用する場合の、場所の指定を行うブロックです。次の変数を指定することができます。

**posx** 指定したい場所の,  $x$  座標を長さの単位で指定します。

**posy** 指定したい場所の,  $y$  座標を長さの単位で指定します。

**posz** 指定したい場所の,  $z$  座標を長さの単位で指定します。

**coordination\_number** ブロック 配位数拘束の定義式における  $\kappa, r_c$  の値を指定するブロックです。次の変数を指定することができます。

**kappa\_inv** (60) 式における  $\kappa$  の値を, 長さの単位で指定します。

**kappa** (60) 式における  $\kappa$  の値をの値を, 1/bohr 単位で指定します。kappa\_inv よりも優先されます。

**rcut** (60) 式における  $r_c$  の値を, 長さの単位で指定します。

**center\_of\_mass** ブロック type として **center\_of\_mass** を採用し, かつ反応座標を変化させる場合「変化させる方向」をここで指定します。

**directionx** 変化させる方向の  $x$  座標を指定します。

**directiony** 変化させる方向の  $y$  座標を指定します。

**directionz** 変化させる方向の  $z$  座標を指定します。

拘束条件の指定の次は, 採用するダイナミクスのアルゴリズムを指定します。通常の PHASE の入力と同様, **structure\_evolution** ブロックの下で行います。

```
structure_evolution{
  method=quench
  dt=40
  ...
}
```

ここで, **method** としては **quench**, **damp**, **velocity-verlet**, **temperature\_control** を利用することができます。拘束条件を課している場合, **gdiis**, **cg** などには現バージョンでは未対応なのでご注意ください。また, **damp** は **damped molecular dynamics** 法による構造最適化を実施する場合に指定します。この手法は, 多くの場合単純な **quenched MD** よりは大きな時間刻み (**dt**) を採用することができ, 速く収束させることのできる手法です。

### 17.2.2 計算の実行方法

拘束条件付きダイナミクスは「反応座標を逐次変化させて計算する」ケースにおいては NEB の計算と同様原子配置に対する並列計算に対応しています。PHASE を以下のように起動してください。

```
% mpirun -np NP phase ne=NE nk=NK nr=NR
```

ここで, **NP** が MPI プロセス数, **NE** がバンド並列数, **NK** が **k** 点並列数, **NR** が原子配置並列数であり, **NP = NE x NK x NR** という関係が成立している必要があります。この機能を利用する場合, 継続計算の処理がプログラム内で若干変化するので, 継続計算間で **nr** を指定したりしなかったりするとエラーとなる点にご注意ください。**nr** は 1 でも構わないので, 原子配置並列を継続計算のあるタイミングで無効にする場合, **NR** を 1 とすれば目的の動作を達成することができます。

### 17.2.3 出力ファイル

出力ファイルは、「反応座標を逐次変化させる」機能を利用していない場合は通常の PHASE の出力と同様です。すなわち、file\_names.data ファイルにおいて F\_ENF 識別子によって指定されるファイルに各ステップにおけるエネルギーや原子に働く力の最大値が、F\_DYNM 識別子によって指定されるファイルに各ステップにおける原子配置や各原子に働く力が出力されます。ただし、「原子に働く力の最大値」は、拘束条件を課すために必要な「拘束力」も含む点に注意が必要です。

他方、「反応座標を逐次変化させる」計算を実行している場合、次のようなファイル群が出力されます(ここで、F\_ENF 識別子によって指定されるファイルのファイル名を nfefn.data, F\_DYNM 識別子によって指定されるファイルのファイル名を nfdynm.data とします)。

nfefn.data.reacxx xx 番目の反応座標の、各ステップにおけるエネルギーおよび原子に働く力の最大値のデータが出力されます。

nfefn.data.converged (構造最適化の場合のみ) 各反応座標において、反応座標の値そのものと、収束したエネルギーおよび原子に働く力の最大値が出力されます。反応座標とエネルギーの関係をプロットすることによって、「反応経路」と「エネルギー」の関係を解析することができます。

nfdynm.data.reacxx xx 番目の反応座標の、各ステップにおける原子配置や原子に働く力が出力されます。

nfdynm.data.converged (構造最適化の場合のみ) 各反応座標において、構造最適化が収束した後の原子配置が出力されます。

nfbldmoon.data.reacxx(分子動力学シミュレーションの場合のみ) xx 番目の反応座標の、自由エネルギー差を算出するために必要なラグランジュの未定乗数の値が記録されます。

さらに、継続計算ファイルや波動関数・電荷密度ファイルなどは反応座標ごとに出力されます。

## 17.3 複数の反応座標を逐次変化させる方法

複数の反応座標を逐次変化させる場合のプログラムの振る舞いを説明します。たとえば以下のように入力で指定した場合について説明します。

```
structure{
  ....
  ....
  constrainable1{
    mobile = off
    monitor = on
    type = dihedral_angle
    atom1 = 2
    atom2 = 4
    atom3 = 3
    atom4 = 1
    reaction_coordinate{
      sw_reaction_coordinate = on
      init_value = -179 degree
      final_value = -1 degree
      increment = 5 degree
    }
  }
}
constrainable2{
  type=bond_length
  monitor=on
  atom1=3
  atom2=4
  reaction_coordinate{
    sw_reaction_coordinate=on
```



```

        init_value = 1.2 angstrom
        final_value = 1.6 angstrom
        increment = 0.05 angstrom
    }
}
....
....
}

```

まず, `constrainable1` ブロックにおいて 2 面角を  $-179^\circ$  から  $-1^\circ$  まで  $5^\circ$  刻みで変化させるように指定しています。さらに, `constrainable2` ブロックにおいてはボンド長を  $1.2 \text{ \AA}$  から  $1.6 \text{ \AA}$  まで  $0.05 \text{ \AA}$  刻みで変化させるように指定をしています。このような入力を記述した場合, まずボンド長を  $1.2 \text{ \AA}$  に固定した状態で 2 面角を  $-179^\circ$  から  $-1^\circ$  まで変化させて計算が実行されます。 $-1^\circ$  の計算が終了したら, 次はボンド長を  $1.25 \text{ \AA}$  に変化させ, 今度は  $-1^\circ$  から  $-179^\circ$  まで 2 面角を変化させる計算を実行します。このような変化のさせ方を採用することによって, 隣り合う反応座標の組の間で原子配置が極端に変化することを防いでいます。

以上のような方針で検討する反応座標が決まりますので, 合計すると検討する反応座標の数は反応座標  $\alpha$  において検討する反応座標の数を  $n_\alpha$  とすると  $\prod_\alpha n_\alpha$  となります。これよりもきめ細やかに反応座標の組を指定するには, 次に説明する「反応座標の変化の仕方をファイルを通して指定する」機能を利用します。

#### 17.4 ファイルから反応座標の変化の仕方を指定する方法

拘束条件の変化のさせ方は上述の `reaction_coordinate` ブロックにおいて指定しますが, この方法の場合は等間隔の指定です。特に前述の, 複数の反応座標を変化させる計算においては, 反応座標  $\alpha$  において検討する反応座標の数を  $n_\alpha$  とすると  $\prod_\alpha n_\alpha$  個の反応座標を検討することになり, 計算時間が膨大になることがあります。このような制限が問題となる場合に, 「反応座標 (の組)」をファイルから指定することが可能となっています。

まず, 検討したい拘束条件を `constrainablexx` ブロックにおいて通常通り指定します。次に, `structure` ブロックの下に以下の変数を定義します。

```

structure{
    ....
    reac_coord_generation = via_file
    ....
}

```

最後に, 作業ディレクトリーに `reac_coords.data` というファイル名のファイルを作成し, 次のような内容を記述します。

1	-1.9373154697	2.2676711906
2	-1.7627825445	2.2676711906
3	-1.5882496193	2.2676711906
4	-1.4137166941	2.2676711906
5	-1.2391837689	2.2676711906
6	-1.0646508437	2.2676711906
7	-0.8901179185	2.2676711906
8	-0.7155849933	2.2676711906
9	-0.7155849933	2.3621574902
10	-0.8901179185	2.3621574902
11	-1.0646508437	2.3621574902
12	-1.2391837689	2.3621574902
13	-1.4137166941	2.3621574902
14	-1.5882496193	2.3621574902
15	-1.7627825445	2.3621574902
16	-1.9373154697	2.3621574902
17	-1.9373154697	2.4566437898
18	-1.7627825445	2.4566437898
19	-1.5882496193	2.4566437898
20	-1.4137166941	2.4566437898

21	-1.2391837689	2.4566437898
22	-1.0646508437	2.4566437898
23	-0.8901179185	2.4566437898
24	-0.7155849933	2.4566437898
	.....	
	.....	
	.....	

各行が 1 つの反応座標の組に相当します。行の 1 列目にはその反応座標の組を識別するための番号を整数で入力します。2 列目以降は、定義した拘束条件の順番で反応座標の値を入力します。この例では 2 種類の反応座標を検討していることになります。1 つ目の反応座標の組では 1 番目の拘束条件として “-1.9373154697” という値、2 番目の拘束条件として “2.26711906” という値を指定することになります。単位は、PHASE のデフォルトの単位を利用して指定するようにしてください。長さならば bohr 単位、角度なら radian 単位です。

### 17.5 Blue Moon 法によって自由エネルギーを計算する方法

拘束条件付きの分子動力学シミュレーションが発生する統計集合 (blue moon ensemble) のデータを利用すると、検討した反応座標の経路上における自由エネルギーの変化を算出することができます [9]。

反応座標が  $\xi_1$  から  $\xi_2$  へ変化する場合の自由エネルギー差は、次のように計算することが可能です。

$$W(\xi_1) - W(\xi_2) = \int_{\xi_2}^{\xi_1} d\xi \frac{\partial W}{\partial \xi}. \quad (61)$$

ここで自由エネルギーの反応座標微分、 $\left\langle \frac{\partial W}{\partial \xi} \right\rangle_{\xi}$  は mean force と呼ばれる物理量であり、ハミルトニアン  $H$  の反応座標微分と次のような関係があります。

$$\frac{\partial W}{\partial \xi} = \left\langle \frac{\partial H}{\partial \xi} \right\rangle_{\xi}^{\text{cond}}. \quad (62)$$

ここで  $\langle \dots \rangle^{\text{cond}}$  とは「条件付き統計平均」です。拘束条件付き分子動力学シミュレーションの統計平均と条件付き統計平均は単純には結びつきませんが、(62) 式は拘束条件付き分子動力学を遂行する際に計算するラグランジュの未定乗数  $\lambda$  を利用して以下のように計算することができます。

$$\begin{aligned} \frac{\partial W}{\partial \xi} &= - \frac{\left\langle |\Xi|^{-1/2} \lambda \right\rangle}{\left\langle |\Xi|^{-1/2} \right\rangle} \\ \Xi &= \sum_i \frac{1}{m_i} \frac{\partial \xi}{\partial \vec{r}_i} \frac{\partial \xi}{\partial \vec{r}_i} \end{aligned} \quad (63)$$

(63) 式には、厳密にはより複雑な補正項が付きませんが、実用上は問題ないとされています。

PHASE による拘束条件付き分子動力学シミュレーションの結果から (61) 式、(63) 式を利用して自由エネルギー差を計算するには、PHASE パッケージに付属している blumoon プログラムを利用します<sup>1</sup>。

blumoon プログラムのソースコードは、PHASE インストールディレクトリーの src\_bm ディレクトリーに納められています。blumoon プログラムは Fortran90 コンパイラーと C コンパイラーを必要とします。Fortran90 コンパイラーを環境変数 F90 に、C コンパイラーを環境変数 CC に設定し、make コマンドを発行すればコンパイルすることができます。以下はお使いのシステムが bash で、Fortran コンパイラーのコマンドが f90、C コンパイラーのコマンドが cc の場合の例です。

```
% cd phase_v1000
% cd src_bm
% export F90=f90
% export CC=cc
% make
% make install
```

<sup>1</sup> 現バージョンでは、blumoon プログラムは反応座標が 1 つの場合のみに対応しています。

環境変数 F90 と CC の指定がない場合, gfortran と gcc がデフォルト値として利用されます。コンパイルが終了すると, bluemoon という名前の小さなプログラムが作成されます。% make install とすると phase\_v1000/bin ディレクトリーの下に bluemoon を移すことができます。

bluemoon プログラムの入力ファイルは, PHASE のそれと同等です。nfnnp.data ファイルに thermodynamic\_integration ブロックを作成し, 計算条件を入力します。たとえば以下ようになります。

```
thermodynamic_integration{
  nsteps=2000
  nequib=1000
  istart_reac_coords=1
  nreac_coords=14
  nsample=10
  smooth=off
  basedir=.
}
```

thermodynamic\_integration ブロックでは以下の指定を行うことができます。

**nsteps** 各反応座標における分子動力学シミュレーションの総ステップ数を指定します。デフォルト値は 2000 ですが, 実施した計算に合わせて変更してください。

**nequib** nsteps の内, 平衡化のため捨てるステップ数を指定します。nsteps よりも小さく, 熱平衡に至ったと考えられる値を指定してください。

**istart\_reac\_coords** 最初に検討する反応座標の ID を入力します。デフォルト値は 1 です。

**nreac\_coords** 最後に検討する反応座標の数を指定します。

**nsample** 統計誤差を見積もる場合にシミュレーションを何分割するかを指定します。

**smooth** on とすると, 三次のスプライン関数によって計算結果を滑らかにします。

**basedir** 結果を出力するディレクトリーを指定します。デフォルト値はカレントディレクトリーです。

以上のような入力を作成したら, 次のように bluemoon を走らせます。

```
% bluemoon inputfile
```

引数で指定する inputfile は入力ファイルのファイル名です。指定がない場合, nfnnp.data という文字列が採用されます。

この計算はさほど時間がかからないはずで, 計算が終了すると, 次のファイルが作成されます。

**potential\_of\_mean\_force.data** 自由エネルギーの計算結果が出力されます。以下のような形式で出力されます。

```
#value, potential of mean force in Hartree, eV, kcal/mol, kJ/mol
2.4566437898 -0.0215821952 0.0003443042 -0.5872816633 0.0093689992 -13.5430301648 0.2160541460 -56.6640534911 0.9039707906
2.2676711910 -0.0224669448 0.0003796767 -0.6113569350 0.0103315334 -14.0982188431 0.2382507016 -58.9869635475 0.9968412043
2.0786985910 -0.0226882285 0.0004435350 -0.6173783747 0.0120692073 -14.2370764737 0.2783223931 -59.5679440305 1.1645012069
.....
.....
```

各行が 1 つの反応座標のデータに相当します。1 列目が反応座標の値, 2 列目, 3 列目がハートリー単位, 4 列目, 5 列目が電子ボルト単位, 6 行目と 7 行目が kcal/mol 単位, 8 行目と 9 行目が kJ/mol 単位での自由エネルギーとその統計誤差の結果に対応します。

**mean\_force\_raw.data** 検討した反応座標から得られる mean force の計算結果が出力されます。次のような形式で出力されます。

```
2.4566437898 0.0066082098 0.0188118786
2.2676711910 0.0034758686 0.0099291734
2.0786985910 -0.0009537509 0.0028573953
1.8897259920 -0.0074922663 0.0213420952
1.7007533930 -0.0098143395 0.0279585555
1.5117807940 -0.0157974842 0.0449758051
1.3228081950 -0.0161451965 0.0459534340
.....
.....
```

potential\_of\_mean\_force.data ファイルと同様に、各行が 1 つの反応座標のデータに相当します。1 列目が反応座標の値、2 列目が mean force の値 (単位: hartree/対応する反応座標の単位)、3 列目が統計誤差に相当します。

mean\_force\_smoothed.data 三次のスプライン関数によって自由エネルギー計算を滑らかにする場合 mean force を滑らかにしたあとに (61) 式の積分を実施しますが、その滑らかにした mean force の計算結果が出力されます。そのデータ形式は、mean\_force\_raw.data ファイルから統計誤差の列を除いたものになります。

## 17.6 例題

拘束条件付き構造最適化計算の例として、 $\text{H}_2\text{O}_2$  および  $\text{H}_2\text{S}_2$  分子の回転障壁の解析例を紹介します。 $\text{H}_2\text{O}_2$ ,  $\text{H}_2\text{S}_2$  は図 29 で示す分子構造を有する単純な分子です。HOOH (HSSH) が成す 2 面角の回転ポテンシャルは、H 原子同士の相互作用と H 原子と O(S) 原子の孤立原子対との相互作用が競合し、W 型ポテンシャルになることが知られています。2 面角を拘束した構造最適化を複数の 2 面角において実施することにより、このような振る舞いが得られるかどうかを確認します。

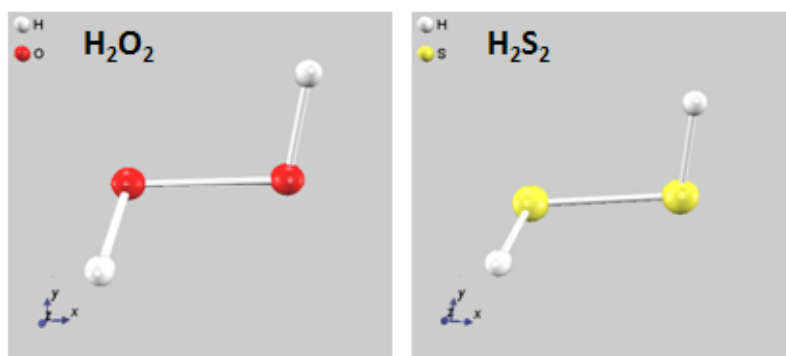


図 29:  $\text{H}_2\text{O}_2$  および  $\text{H}_2\text{S}_2$  分子の分子構造。

この例題の入力ファイルは、samples ディレクトリーの下で constraints 以下、 $\text{H}_2\text{O}_2$  および  $\text{H}_2\text{S}_2$  ディレクトリーにあります。まず、structure ブロックの下には以下の記述があります。

```
structure{
  constrainable1{
    type = dihedral_angle
    atom1 = 2
    atom2 = 4
    atom3 = 3
    atom4 = 1
    reaction_coordinate{
      sw_reaction_coordinate = on
      init_value = 9 degree
      final_value = 179 degree
      increment = 10 degree
    }
  }
  ...
  ...
}
```

constrainable1 ブロックを作成し、その下で拘束条件の指定を行っています。この例題では拘束条件は一つのみ課しますが、互いに相いれる拘束条件であるならばいくつでも定義することが可能です。今は 2 面角の拘束を実施するので、type 変数としては dihedral\_angle を指定しています。また、2 面角を定義するために必要な 4 つの原子の番号を atom1 から atom4 変数によって指定しています。さらに、reaction\_coordinate ブロックを作成し、この拘束条件を逐次変化させる指定を行います。sw\_reaction\_coordinate を on、init\_value と final\_value をそれぞれ 9 degree と 179 degree、increment を 10 degree としていますが、このような指定によって、9 °から 179 °まで、10 °刻みで 2 面角を変化させて構造最適化を行います。

図 30 に、2 面角と最適化の結果得られたエネルギーの関係を示します。図 30 には、実験結果 [8] も併せて実線で表示しています。一見して明らかなように、計算結果と実験結果はよい一致が得られています（おおそ 1 kcal/mol 程度の違い）。

$\text{H}_2\text{O}_2$  と  $\text{H}_2\text{S}_2$  の大きな違いは 2 点あります。1 点目は、安定な 2 面角の値です。 $\text{H}_2\text{O}_2$  は 4 面体の角度である  $109.5^\circ$  に近い値が安定であるのに対し、 $\text{H}_2\text{S}_2$  は  $90^\circ$  付近が安定な 2 面角です。2 点目は trans 障壁エネルギー（図 30 では  $180^\circ$  付近の障壁エネルギー）の高さです。 $\text{H}_2\text{O}_2$  と比較すると、 $\text{H}_2\text{S}_2$  の trans 障壁ははるかに大きく、実験的には約 6 倍の値が得られています。いずれの点も本計算によって再現されており、妥当な結果が得られているものと考えられます。

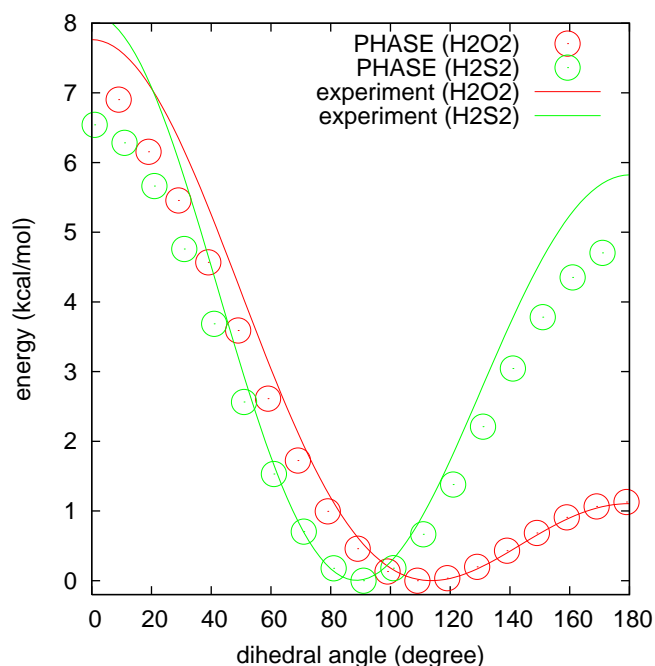


図 30:  $\text{H}_2\text{O}_2$  および  $\text{H}_2\text{S}_2$  分子の 2 面角とエネルギーの関係。

## 18 Meta-dynamics 法

### 18.1 はじめに

Meta-dynamics 法 [10,11] は、化学反応などの障壁エネルギーの存在する過程を効率よく解析するための手法です。Meta-dynamics 法においては、 $S_\alpha(r)$  という“集団変数”を導入します。ここでいう集団変数とは、具体的には対象とする系の原子座標から定義可能な反応座標（ボンド長やボンド角などの内部座標や配位数など）を複数集めたものです。各集団変数には、仮想的な“粒子”が割り当てられます。この、“仮想的な粒子の運動”のことを Meta-dynamics とよびます。Meta-dynamics のアルゴリズムをうまく設計することによって、効率よく（検討している集団変数が作る）自由エネルギー表面を探索することができると考えられます。ここでは、PHASE に実装された Meta-dynamics 法の利用方法を説明します。

### 18.2 理論

#### 18.2.1 概要

Meta-dynamics 法では、計算の履歴に依存するバイアスポテンシャル  $V(t, s)$  をある間隔（通常数十から数百 MD ステップ）で足しこんでいきます。このような方針を採用することによって、自由エネルギー空間において一度訪れた点に訪れづらくする効果が発揮されます。十分長い時間シミュレーションを行うと  $V(t, s)$  が自由エネルギー空間を埋め尽くしてしまい、反応は自由に起こることができるようになります。この状態に至る  $V(t, s)$ （に-1を掛けた量）がすなわち自由エネルギーであるとみなすことができます。

Meta-dynamics 法によるシミュレーションの模式的な様子を図 31 に示します。この図では、まずシミュレーションは 1 の数字が割り当てられた谷から始まります。2 のバイアスを足し、さらに 3 のバイアスを足すと新しい局所極小（図中で最も左側の谷）に至ります。さらに 4, 5, 6 とバイアスポテンシャルを足すともっともエネルギーの低い谷（図中で最も右側の谷）へ至ることができます。最後に 7 のバイアスを足し、さらに 8 のバイアスポテンシャルまで足すと、系は集団変数の空間を自由に行き来できるようになります。この時点でのバイアスポテンシャルに-1を掛けると、それは自由エネルギーと見做すことができると分かります。

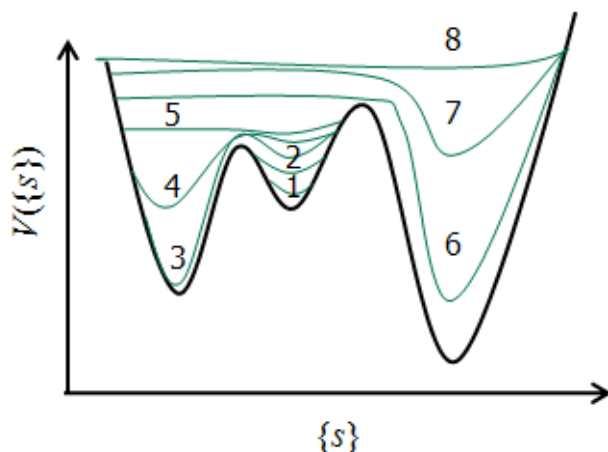


図 31: Meta-dynamics シミュレーションの模式図。

Meta-dynamics 法の特徴として、反応座標と関連付けられた複数の動力学変数の動力学を追跡する点が挙げられます。この考え方を導入することによって、複数の反応座標を検討することが容易となり、また動力学変数自身が“もっともらしい”反応経路を探索してくれる効果が期待できます。また、blue moon 法の場合複数の反応座標を取り扱うことは（原理上は可能ではありますが）難しいのに対し、meta dynamics 法においては比較的容易です。したがって、反応座標が複数ある場合や、反応経路が自明でない場合などにおいて有効な方法であると考えられます。バイアスポテンシャル導入の方針の違いにより、自由エネルギー表面を粗く、すばやく探索することも、きめ細かく、精度よく探索することも可能です。

Meta-dynamics 法シミュレーションは、あらかじめ決められた間隔でバイアスポテンシャルを足しながら進行していきます。この際にバイアスポテンシャルを構築するには、時刻 0 から現在までのデータをすべて利用して

和を取る必要があるので、Meta-dynamics 法は  $O(t^2)$  の計算手法となります（ただし、第一原理計算で利用する  
かぎりこの点が制約になることはないでしょう）。

### 18.2.2 運動方程式のもととなるハミルトニアン

Meta-dynamics 法のハミルトニアンは、具体的には、次のように記述されます。

$$H_{\text{meta}} = H_{\text{MD}} + \sum_{\alpha} \frac{1}{2} \mu_{\alpha} \dot{s}_{\alpha}^2 + \sum_{\alpha} \frac{1}{2} k_{\alpha} (S_{\alpha}(\mathbf{r}) - s_{\alpha})^2 + V(t, s),$$

$$V(t, s) = \sum_{t_i < t} w \exp \left[ - \sum_{\alpha} \frac{(s_{\alpha}(t) - s(t_i))^2}{2 \Delta s_{\alpha}^2} \right] \quad (64)$$

ここで、 $\alpha$  は集団変数に含まれる各変数を識別する変数、 $\mu_{\alpha}$  と  $s_{\alpha}$  はそれぞれ仮想的な粒子の質量と座標、 $S_{\alpha}(\mathbf{r})$  は対象としているシステムから定義される「集団変数」、 $k_{\alpha}$  は仮想的な粒子の座標と集団変数を結びつける「ばね定数」、 $V(t, s)$  がバイアスポテンシャルです。足しこんでいったバイアスポテンシャルを記録しておく、そこから自由エネルギーを見積もることも可能です。このようなハミルトニアンから得られる動力学は、次のようにまとめることができます。

- 系は、集団変数を通して仮想的な粒子の座標値に緩く拘束される。
- 仮想的な粒子の座標は、バイアスポテンシャルの効果によって、すでに訪れた点には再訪づらい。

仮想的な粒子の座標の運動と系の運動の特徴的なタイムスケールが異なれば（仮想的な粒子の方が長いタイムスケールであれば）、系の運動は仮想的な粒子の運動の影響をそれほどは受けないので局所的には正しく系の運動を追跡し、かつゆるやかに集団変数の張る空間を探索することが可能となります。仮想的な粒子の質量は、上記の原理より集団変数の固有振動モードが系よりも遅くなるように設定します。

Meta-dynamics 法は、仮想粒子の動力学を追跡するのではなく、系に直接バイアスポテンシャルを足しこんでいくことによって実現する手法もあります [12]。このような方針を採用すると、仮想的な粒子の質量やそれと集団変数を結びつけるばね定数の定義が不要となり、よりシンプルに実行することが可能となります。

## 18.3 利用方法

### 18.3.1 入力ファイルの記述

Meta-dynamics 法解析プログラムを利用するためには、次の設定が必要です。

- PHASE に Meta-dynamics 法を利用することを伝える
- Meta-dynamics の振る舞いを規定する（ダイナミクス追跡モード、バイアスポテンシャル更新回数、出力形式など）
- 温度一定の MD シミュレーションの設定を行う
- 集団変数の定義を行う（集団変数に含める反応座標の情報を集団変数の数だけ定義）
- バイアスポテンシャルの設定を行う（バイアスポテンシャルの高さ、幅、更新頻度など）
- レプリカ並列計算を実行する場合、その設定。

以下、各々について詳しく説明します。なお、PHASE 入力ファイルの基本的な形式や通常の計算設定と共通する設定（計算モデルの指定、波動関数ソルバーの指定など）はここでは説明しないので、これらについては本ユーザーマニュアルの該当箇所をご覧ください。

PHASE に Meta-dynamics 法を利用することを伝える PHASE に Meta-dynamics 法を利用することを伝えるには、control ブロックにおいて以下の指定を行います

```
control{
    driver = meta_dynamics
}
```

この指定により、PHASE の通常の原子ダイナミクスコードではなく、Meta-dynamics 追跡用のメイン・プログラムが呼ばれます。

**Meta-dynamics の振る舞いの設定** Meta-dynamics の全体的な振る舞いの設定方法を説明します。この設定は、入力ファイルの最上位に meta\_dynamics ブロックを作成し、さらに以下のような変数・ブロックを定義することによって行います。

```
meta_dynamics{
  meta_dynamics_type = bias_only
  max_bias_update = -1
  extensive_output=on
  output_per_rank=on
  output_cvar_every_step=off
  continuation_strategy{
    randomize_velocity=on
    scale_velocity=off
    velocity_scaling_factor=0.7
    configuration_from_input=off
    ...
  }
}
```

meta\_dynamics ブロックでは、以下の変数/ブロックの設定を行うことができます。

**meta\_dynamics\_type** 変数 bias\_and\_fictitious, bias\_only, bias\_generation のいずれかを指定します。bias\_and\_fictitious とすると仮想粒子の動力学を考慮したメタダイナミクスを、bias\_only とするとバイアスポテンシャルのみを有効にしたメタダイナミクスを実行することができます。bias\_generation とした場合 Meta dynamics は実行されず、作業ディレクトリーに存在するファイルからバイアスポテンシャルの構築および出力のみが実行されます。

**max\_bias\_update** 変数 最大何回バイアスポテンシャルを更新するかを指定します。負の値を指定すると、バイアスポテンシャルの更新回数では計算は停止しません。これがデフォルトの振る舞いです。

**output\_per\_rank** 変数 on とすると、レプリカ並列計算実行時に出力が各ランクごとに得られます。

**extensive\_output** 変数 on とすると、仮想粒子の速度や仮想粒子に働く力など、通常は不要な出力も得られます。

**continuation\_strategy** ブロック レプリカ並列計算実行時における継続計算の方針の設定を行います。レプリカ並列計算時には、並列数を変化させた場合に以前の計算を厳密に再現することはできないので、ここでどのような方針で継続するのか決定する必要があります。このブロックでは、以下の設定を行うことができます。

**randomize\_velocity** on とすると、継続計算実行時に速度を継続せず、乱数で割り振ります。デフォルト値は off。

**scale\_velocity** on とすると、読みこんだ速度をつぎに説明する velocity\_scaling\_factor の値に応じてスケールします。デフォルト値は off。

**velocity\_scaling\_factor** 読み込んだ速度にここで指定した値をかけます。デフォルト値は 1。

**configuration\_from\_input** on とすると、継続計算ファイルではなく入力ファイルから座標データを読み込む。デフォルト値は off。

**集団変数の定義** 集団変数とは、“反応座標”を複数まとめたものです。この指定は、meta\_dynamics ブロック以下において行います。典型的な例は下記の通りです。

```
meta_dynamics{
  ....
  ....
  collective_variable{
    mass=1000
    k=100
    delta_s = 0.08
    control_velocity=on
    mass_thermo = 50
  }
}
```



```

    target_KE = 0.1
}
collective_variable1{
    type=bond_length
    atom1=5
    atom2=4
    delta_s=0.05 angstrom
    smin=1 angstrom
    smax=5 angstrom
    ds = 0.1 angstrom
}
....
....
}

```

まず, meta\_dynamics ブロックの下に collective\_variable ブロックを作成します。collective\_variable ブロックには全集団変数に共通の設定を行います。後に説明する集団変数個別の設定に記述がない項目のみここでの設定が反映されます。

次に, 集団変数を定義したい数だけ collective\_variable $xx$  ブロックにおいて定義します。ここで  $xx$  は集団変数の ID です。任意の数の集団変数を定義することが可能ですが, 1 から連続的に変化する整数を指定する必要があります。たとえば, collective\_variable1, collective\_variable2, collective\_variable4 の 3 つの collective\_variable $xx$  ブロックがあった場合, collective\_variable1 と collective\_variable2 のみ解釈されます。

collective\_variable および collective\_variable $xx$  ブロックは, 拘束条件付きダイナミクスの設定の際に設定する拘束条件と同様の変数を定義することが可能となっています。具体的には, 以下の変数を定義することが可能です。

**type** 変数 集団変数の“種類”を指定します。以下のいずれかの値をとります。

bond\_length 2 原子間の距離を集団変数とします。  
bond\_angle 3 原子の成すボンド角を集団変数とします。  
dihedral\_angle 4 原子の 2 面角を集団変数とします。  
bond\_length\_diff 2 原子間の距離の差を集団変数とします。  
plane ある原子の指定の面内での位置集団変数とします。  
center\_of\_mass 指定の原子群の重心を位置集団変数とします。  
coordination\_number 配位数を位置集団変数とします。  
distance\_from\_pos ある場所からの距離を集団変数とします。

**atomx** 変数 指定の集団変数に関わる原子を指定します。x は数字であり, たとえば 2 原子間の距離の場合は 2 つの原子が拘束に関わるので, atom1 と atom2 に対応する原子の番号を指定します。type が coordination\_number の場合, 配位数を計算する中心の原子の番号を指定します。

**plane** ブロック 面内拘束の場合の, 拘束したい面の法線ベクトルを指定するためのブロックです。次の変数を指定することができます。

normx 法線ベクトルの x 座標。  
normy 法線ベクトルの y 座標。  
normz 法線ベクトルの z 座標。

**coordination\_number** ブロック 配位数拘束の場合原子  $i$  の配位数  $n_i$  は  $n_i = \sum_{j \neq i} \frac{1}{\exp[\kappa(|\mathbf{r}_j - \mathbf{r}_i| - r_c)] + 1}$  と定義していますが, この式における  $\kappa, r_c$  の値を指定するブロックです。次の変数を指定することができます。

kappa\_inv  $\frac{1}{\kappa}$  の値を, 長さの単位で指定します。  
kappa  $\kappa$  の値を, 1/bohr 単位で指定します。kappa\_inv よりも優先されます。  
rcut  $r_c$  の値を長さの単位で指定します。

**mass** 変数 仮想粒子の質量を指定します。meta\_dynamics\_type が bias\_and\_fictitious の場合のみ意味のある指定です。

**k** 変数 仮想粒子と集団変数の結びつきを決める,ばね定数を指定します。meta\_dynamics\_type が bias\_and\_fictitious の場合のみ意味のある指定です。

**delta\_s** 変数 (64) 式の  $\delta s_\alpha$  の値を指定します。

**smin** 変数 バイアスポテンシャル出力の際の最小値を指定します。

**smax** 変数 バイアスポテンシャル出力の際の最大値を指定します。

**ds** 変数 バイアスポテンシャル出力の際の“ 刻み幅 ”を指定します。

**control\_velocity** 変数 “ on ”にした場合,仮想粒子のダイナミクスを追跡する際に熱浴を付与することによってその速度を制御します。meta\_dynamics\_type が bias\_and\_fictitious の場合のみ意味のある指定です。

**mass\_thermo** 変数 control\_velocity が“ on ”の場合の,熱浴の“ 質量 ”を指定します。

**target\_KE** 変数 control\_velocity が“ on ”の場合の,目的とする仮想粒子の温度を指定する。

**バイアスポテンシャルの設定** バイアスポテンシャルの設定は, meta\_dynamics ブロックの下に bias\_potential ブロックを作成して行います。以下に典型的な例を示します。

```
bias_potential{
  height = 0.02 eV
  update_frequency=20
  output_frequency=100
}
```

bias\_potential ブロックにおいて定義可能な変数は下記の通りです。

**height** 変数 一度に足すバイアスポテンシャルの高さをエネルギーの単位で指定します。一度に足すバイアスポテンシャルの幅は,各集団変数固有の量であるので bias\_potential ブロックではなく集団変数固有の設定を行う collective\_variablexx ブロック以下で行います。

**output\_frequency** 変数 meta\_dynamics\_type が bias\_generation の場合のみ意味のある指定です。何回に 1 回バイアスポテンシャルを出力するかを指定します。

**update\_frequency** バイアスポテンシャルの更新頻度を指定します。デフォルト値は 20 です。

**レプリカ並列計算の設定** 複数のレプリカを導入した並列計算を行う際は,以下の 2 つの方法で初期座標・初期速度を設定します。

#### 1. 初期速度を変化させる

特に指定がない場合,初期座標はすべてのレプリカで共通で,初期速度の乱数のみ異なる,という条件で計算がなされます。位相空間上異なる点から始めるので,座標が同じでも各レプリカはいずれ異なる軌跡をとるようになります。ただし,当然のことながら最初のうちは(座標値は)ほぼ同じ軌跡となります。

#### 2. 初期の座標値をランクごとに明示的に指定する

入力ファイルにおいて,レプリカごとに異なる座標データを指定することも可能です。この設定は, atomsxx ブロック(ここで xx は MPI ランクの数字)を作成し,そこで座標値を設定することによって行います。たとえば,ランク 0 が担当するレプリカとランク 1 が担当するレプリカにそれぞれ異なる座標値を与えるには,次のような記述を行います。

```
structure{
  atom_list{
    ....
    atoms0{
      #units angstrom
      #default weight = 1, element = Si, mobile = 1
      #tag element rx ry rz mobile weight
      C 5.0157363043      5.6563796505      5.8043454319 1 1
```

```

        C 4.7499007526          4.2727134018          5.7364572058 1 1
        ...
        ...
    }
    atoms1{
        #units angstrom
        #default weight = 1, element = Si, mobile = 1
        #tag element rx ry rz mobile weight
        C      4.5897384578      5.5998560107      5.7723226564 1 1
        C      5.1658344359      4.3217914066      5.6857269157 1 1
        ...
        ...
    }
}
}
}

```

### 18.3.2 計算の実行方法

Meta-dynamics 法を実行するには、通常の PHASE による計算と同様に以下のコマンドを発行します。

```
mpirun -n NP phase ne=NE nk=NK nr=NR
```

ここで NP は MPI プロセス数, NE はバンド並列数, NK は k 点並列数, NR はレプリカ並列数です。NP=NE × NK × NR という関係が成立している必要があります。ne, nk, nr はいずれも省略可能（デフォルト値は 1 すべて無指定の場合 NE=NP）です。

通常 Meta dynamics 実行時に得られるバイアスポテンシャルの出力は「最も最近得られたバイアスポテンシャル」のみですが、バイアスポテンシャルをポスト处理的に計算し、それを出力させることもできます。この機能を利用するには、入力ファイルの meta\_dynamics ブロックの meta\_dynamics\_type 変数に bias\_generation という文字列を指定します。この時、meta\_dynamics ブロックの下に bias\_potential ブロックにおいて定義される、bias\_output\_frequency 変数に指定された回数に 1 回出力を行います。たとえば、bias\_output\_frequency が 10, バイアスの総更新回数が 100 だった場合、10 回目, 20 回目, 30 回目, ... 100 回目の更新時のバイアスポテンシャルがそれぞれ独立したファイルに出力されます。そのファイル名は、" bias\_potential.dataxx " となります。ここで xx が対応する更新回数です。この設定を行ったあと、Meta-dynamics 解析を行ったディレクトリーにおいて PHASE を実行します。ファイルを読み込みバイアスポテンシャルを構築するのみなので、通常並列で実行する必要はありません。

### 18.3.3 出力ファイル

Meta dynamics シミュレーションを行う場合、標準よりも多くのファイルが得られます。以下に、各々について簡単に説明します。

**curr\_bias\_potential.data ファイル** “現在の”バイアスポテンシャルが記録されたファイルです。次のような形式で記録されます。

```

1.2000000000      -3.1400000000      0.0000000000
1.3000000000      -3.1400000000      0.0000000000
1.4000000000      -3.1400000000      0.0000000000
1.5000000000      -3.1400000000      0.0000000000
1.6000000000      -3.1400000000      0.0000000000
1.7000000000      -3.1400000000      0.0000000000
      ....
      ....

1.2000000000      -3.0400000000      0.0000000000

```

1.3000000000	-3.0400000000	0.0000000000
1.4000000000	-3.0400000000	0.0000000000
1.5000000000	-3.0400000000	0.0000000000
1.6000000000	-3.0400000000	0.0000000000
1.7000000000	-3.0400000000	0.0000000000
....		
....		

各行が“ 集団変数の組 ”に相当します。定義している数だけ集団変数が記録されたあと、その“ 集団変数の組 ”におけるバイアスポテンシャルの値が記録されます。

**bias\_potential.data<sub>xx</sub>** ファイル バイアスポテンシャルを作成するのみのモードを利用した場合に得られる、更新回数に応じたバイアスポテンシャルのデータが記録されるファイルです。ファイル名の **xx** がバイアスポテンシャル更新回数に相当します。そのファイル形式は、**curr\_bias\_potential.data** と同様です。

**nfdynm.data\_at\_bias** ファイル バイアスポテンシャル更新時における座標データを記録したファイルです。PHASE の標準座標データ出力形式である、**F\_DYNM** 形式で記録されます。

**nfe<sub>fn</sub>.data\_at\_bias** ファイル バイアスポテンシャル更新時におけるエネルギーの値を記録したファイルです。PHASE の標準的なエネルギーデータ出力形式である、**F\_ENF** 形式で記録されます。

**collective\_variables.data** ファイル バイアスポテンシャル更新時における集団変数の値を記録したファイルです。次のような形式で出力されます。

2	1.6399047278	0.0906233310
3	1.6933783940	0.2327954221
4	1.6487636847	0.0655806009
5	1.7510381463	-0.1403803460
6	1.7880912692	-0.2122517967
7	1.7558411086	-0.2557274737
8	1.7939362867	-0.0296094373
9	1.7595919709	0.1959354384
10	1.7773637731	0.3761827029
11	1.7657919080	0.3998392061
12	1.7604309483	-0.0107912799
13	1.6218441177	-0.3366407543
....		
....		

各行がバイアスポテンシャル更新のタイミングに対応します。一列目がバイアスポテンシャルの更新回数であり、二列目以降定義順に対応する集団変数の値が記録されます。

**bias\_potential.parameters.data** ファイル バイアスポテンシャルのパラメーターが記録されるファイルです。継続計算のタイミングでこのパラメーターを変化させた場合、それ以前のパラメーターの値が分からないとバイアスポテンシャルの構築ができないことから必要なファイルです。次のような形式で出力されます。

2	0.0200000000	0.1000000000	0.1000000000
3	0.0200000000	0.1000000000	0.1000000000
4	0.0200000000	0.1000000000	0.1000000000
5	0.0200000000	0.1000000000	0.1000000000
....			
....			

各行がバイアスポテンシャル更新のタイミングに対応します。一列目がバイアスポテンシャル更新回数であり、二列目が (64) 式における  $w$  の値、3 列以降が各集団変数の (64) 式における  $\delta s_{\alpha}$  の値です。

## 18.4 例題

Meta dynamics 法を利用した例として、炭化水素のエネルギー表面を調べた例を紹介します。具体的には、 $C_4H_6$  分子の電子環状反応を取り上げます。 $C_4H_6$  分子は、trans 1-3 ブタジエン、cis 1-3 ブタジエン、シクロブテンの 3 種類の安定構造が知られています。シクロブテンは環状分子、trans 1-3 ブタジエンは平面状の分子ですが、cis 1-3 ブタジエンは平面状にはならず、2 面角を  $30^\circ$  ほどひねった構造が安定な構造です (gauche 配座)。その分子構造を図 32 に示します。エネルギーは、高い順にシクロブテン、cis 1-3 ブタジエン、trans 1-3 ブタジエンであり、分子の反応としては、1-3 ブタジエンが閉環して環状化合物であるシクロブテンを生成する、あるいは逆にシクロブテンが開環し 1-3 ブタジエンが生成される反応 (電子環状反応)、また、2 種類の 1-3 ブタジエンの間の cis-trans 反応が考えられます。閉環・開環反応は化学結合の切断を要することから大きな障壁エネルギーがあり、1 eV 程度のオーダーであると考えられます。他方、cis から trans への変化はそこまでの障壁はなく、100 meV 程度のオーダーであると考えられます。特に、環状反応においては、1-3 ブタジエンとシクロブテンとでは 2 重結合の数が異なり、電子状態としては全く異なるものであるため、古典的なポテンシャルで取り扱うのは一般に難しいと言えます。この点を PHASE で正しく扱えるかどうかを確認します。

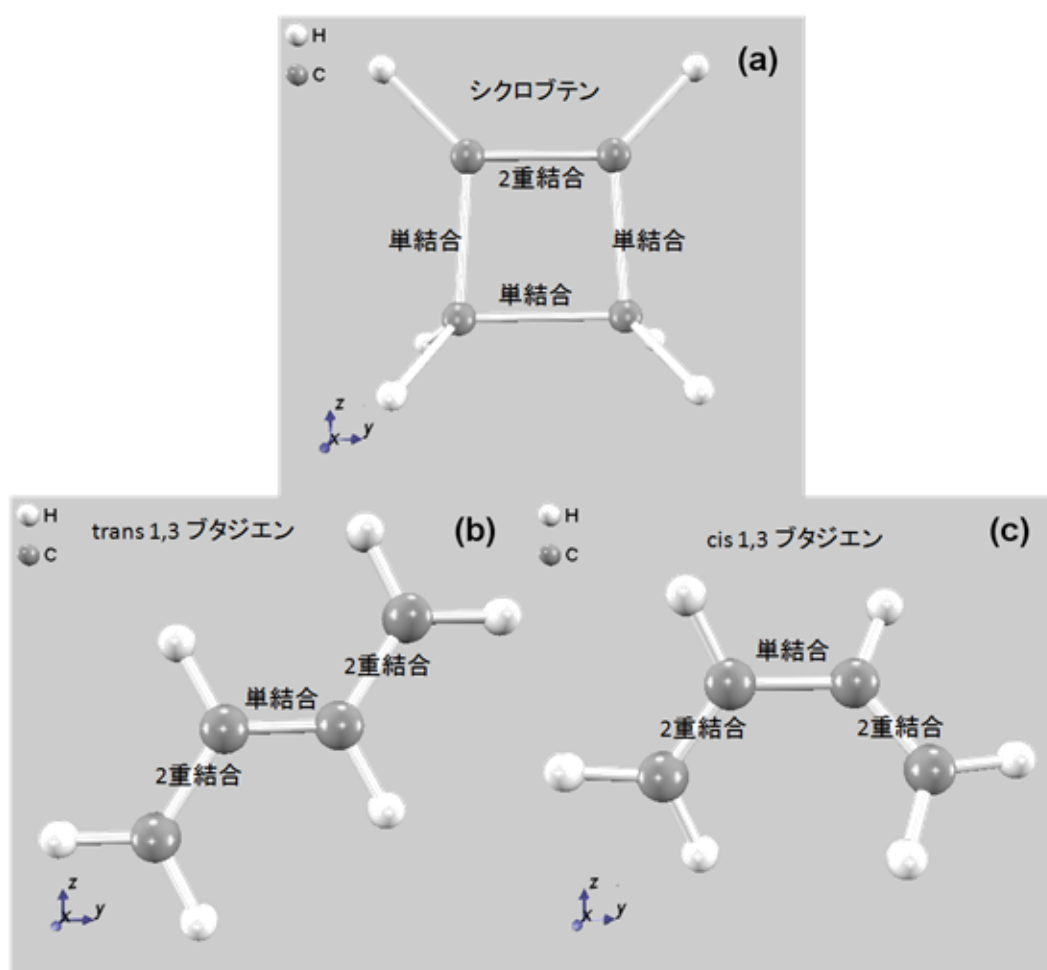
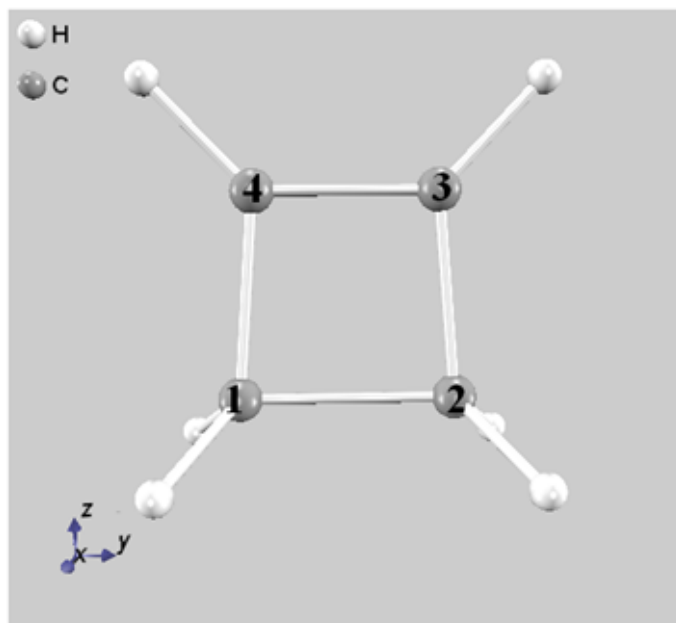


図 32:  $C_4H_6$  分子の分子構造

### 18.4.1 計算方法

初期の原子配置は、図 33 で示すシクロブテンを採用します。この初期構造は、PHASE によって最適化したものです。

まず、プログラムに「Meta dynamics 法を利用したシミュレーションを行う」ことを伝える必要があります。これは、control ブロックの下に driver 変数に meta\_dynamics を指定することによって行います。

図 33:  $C_4H_6$  分子の分子構造

```
condition{
  driver = meta_dynamics
  ....
}
```

次に集団変数を定義します。その方針は様々ですが、ここでは以下を採用します。

1. 図 33 の原子 1 と原子 2 の距離。パラメーター `ds`, `delta_s` はそれぞれ  $0.1 \text{ \AA}$  と  $0.05 \text{ \AA}$
2. 図 33 の, 原子 1-4-3-2 の作る二面角。パラメーター `ds`, `delta_s` はそれぞれ  $10^\circ$  と  $5^\circ$

この設定は, `meta_dynamics` ブロックの下で以下のように実現します。

```
meta_dynamics{
  ....
  ....
  collective_variable1{
    type=bond_length
    atom1=5
    atom2=4
    delta_s=0.05 angstrom
  !for bpot output
    smin=1 angstrom
    smax=5 angstrom
    ds = 0.1 angstrom
  }
  collective_variable2{
    type=dihedral_angle
    atom1=5
    atom2=3
    atom3=2
    atom4=4
    delta_s = 5 degree
  !for bpot output
```

```

    smin = -180 degree
    smax = +180 degree
    ds = 10 degree
  }
}

```

バイアスポテンシャルの高さは 0.02 eV (0.46 kcal/mol) とします。バイアスポテンシャルの更新頻度は、20 MD ステップに一度とします。この設定は、meta\_dynamics ブロックの下に bias\_potential ブロックを作成し、height パラメーターで指定することによって行います。

```

meta_dynamics{
  ....
  ....
  bias_potential{
    update_frequency = 20
    height=0.02 eV
  }
}

```

バイアスポテンシャルを更新する回数は任意ですが、信頼できる自由エネルギー表面を得るためには相当数の更新回数が必要です。

#### 18.4.2 計算結果

本シミュレーションによって得られる計算結果を解説します。まず、図 34 にバイアスポテンシャルを 18,140 回程度更新した結果得られたエネルギー表面の等高線図を示します。

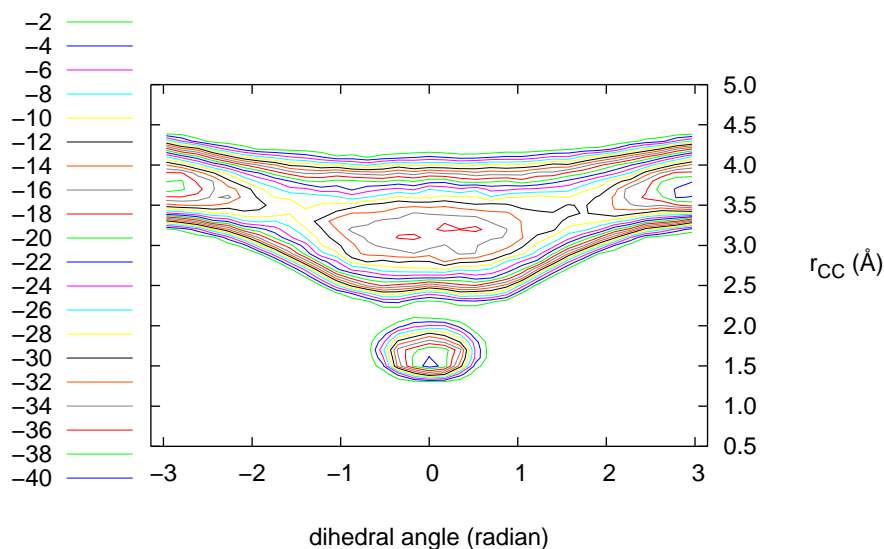


図 34: 本計算例によって得られる  $C_4H_6$  分子の自由エネルギー表面。

図 34 より、ここで得られたエネルギーの等高線図には 4 つの安定点があることが理解できます。すなわち、原子間距離が約 1.5 Å 程度で角度がほぼ 0 radian の点、原子間距離が 3.3 Å 程度で角度が 0 radian、原子間距離が 3.7 Å 程度で角度が  $\pm 3$  radian 程度の 2 つの点です。これらは、それぞれシクロブテン、cis 1-3 ブタジエン、trans 1-3 ブタジエンに相当します。絶対零度の計算の場合、cis ではなく gauche 配座となりますが、300K の Meta

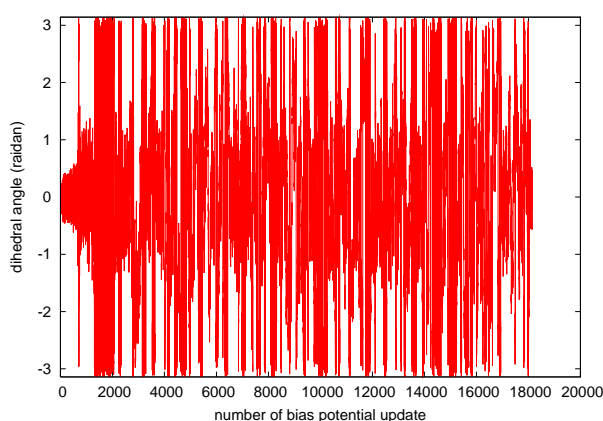


図 35: 2 面角とバイアスポテンシャル更新回数の関係

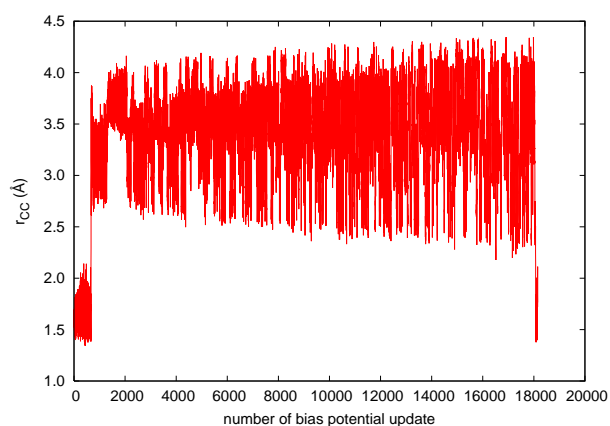


図 36: 炭素原子間距離とバイアスポテンシャル更新回数の関係

dynamics シミュレーションでは *cis* と *gauche* の明確な区別がつけられる結果は得られませんでした。得られたシクロブテンと *trans* 1-3 ブタジエンのエネルギー差は、16 kcal/mol 程度、シクロブテンと *cis* ブタジエンのエネルギー差は 12 kcal/mol 程度となりました。いずれも、絶対零度の計算と比較するとより大きなエネルギー差です。

図 35 および図 36 には、集団変数がバイアスポテンシャルの更新と共にどのように変化していったかを示しています。二面角が図 35、炭素原子間距離が図 36 の振る舞いです。図 35 および図 36 より、バイアスポテンシャルを約 700 回ほど更新した時点で鞍点を越えてブタジエンに至っていることが理解できます。そこから 18,000 回程度の更新までは幅広くエネルギー表面を探索しています。図 32 から分かるように、ここで考えている系はシクロブタンを除くと二面角に対して幅広い範囲の構造を取り得ます。そのため、この谷を埋め尽くすのに多くのバイアスポテンシャルの更新が必要となっています。18,000 回程度のバイアスポテンシャル更新の結果、再びシクロブタンへ戻ったことが確認できた時点(図 36)で計算を終了させました。

図 37 (a) から (d) までに、Meta dynamics シミュレーション中に実際に得られた原子配置のスナップショットを示しました。ここで示しているように、バイアスポテンシャルの効果によって様々な分子構造が実現していることが分かります。



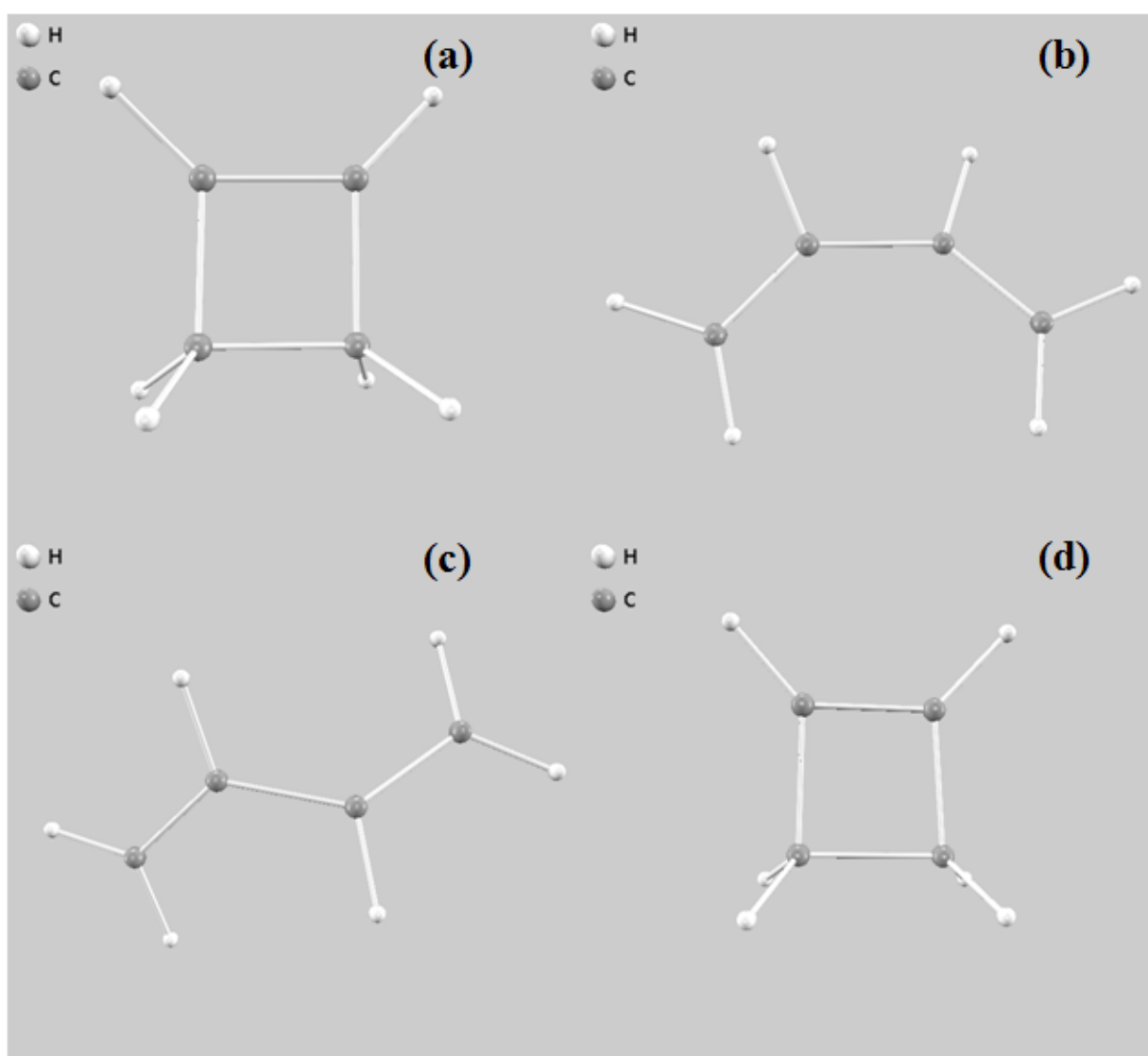


図 37: Meta dynamics シミュレーションによって得られた分子構造のスナップショット . (a) : バイアスポテンシャル 2 回更新 (b) バイアスポテンシャル 690 回更新 (c) バイアスポテンシャル 1,500 回更新 (d) バイアスポテンシャル 18,070 回更新

## 19 非局所相関項

### 19.1 はじめに

本節では非局所相関項を第一原理的に計算する手法 (van der Waals density functional (vdW-DF)) を利用したプログラムについて説明します。PHASE でも採用されている一般化された密度勾配近似: Generalized Gradient Approximation (GGA) では非局所相関項が考慮されていないために、例えば積層グラファイトの層間凝集エネルギーなどを正確に計算することができません。本節で取り上げるルーチンは GGA のこの欠点を補うために用意されたもので、これを用いることで van der Waals 相互作用が大きく寄与する系の全エネルギーや電子状態もより正確に計算できるようになります。また、この vdWDF は第一原理的な手法を利用して経験的なパラメータ等を用いていないので、任意の形状の系に対して簡単に適用することができます。本プログラムは全エネルギーのみを計算する際に用いるポスト計算用のプログラム vdW.F90 と、バンド図などの電子状態も計算する際に用いるセルフ・コンシステント計算用プログラム vc\_nl.F90 の 2 つから構成されます。

### 19.2 非局所相関項の理論 (ポスト計算用)

#### 19.2.1 理論概要

**基本理論** プログラム vdW.F90 では非局所相関項  $E_c^{\text{nl}}$  を計算します。この  $E_c^{\text{nl}}$  に、GGA で得られる交換項  $E_x^{\text{GGA}}$  と LDA から得られる交換項  $E_c^{\text{LDA}}$  を足し合わせることで「非局所相関項も考慮した交換相関項」を導出します。すなわち、交換相関エネルギー  $E_{xc}$  は

$$E_{xc} = E_x^{\text{GGA}} + E_c^{\text{LDA}} + E_c^{\text{nl}} \quad (65)$$

となります。このうち右辺第 3 項の計算が最も困難で、vdW では Dion ら [13] によって開発された理論手法に習って数値計算されます。この理論手法では非局所相関項を

$$E_c^{\text{nl}} = \frac{1}{2} \int d\mathbf{r}_i d\mathbf{r}_k \rho(\mathbf{r}_i) \phi(\mathbf{r}_i, \mathbf{r}_k) \rho(\mathbf{r}_k) \quad (66)$$

として計算します。被積分関数に位置変数が 2 つ ( $\mathbf{r}_i$  と  $\mathbf{r}_k$ ) あるこの式では、GGA や LDA と違って、離れた位置にある電荷密度同士 ( $\rho(\mathbf{r}_i)$  と  $\rho(\mathbf{r}_k)$ ) の相互作用も考慮しています。2 変数関数  $\phi(\mathbf{r}_i, \mathbf{r}_k)$  は

$$\phi(\mathbf{r}_i, \mathbf{r}_k) = \frac{2}{\pi^2} \int_0^\infty da db a^2 b^2 W T \quad (67)$$

のように書けます。ここで、

$$\begin{aligned} W(a, b) &= \frac{2}{a^3 b^3} [(3 - a^2) b \cos b \sin a + (3 - b^2) a \cos a \sin b \\ &+ (a^2 - b^2 - 3) \sin a \sin b - 3ab \cos a \cos b] \end{aligned} \quad (68)$$

です。また、

$$\begin{aligned} T[x_i(a), x_i(b), x_k(a), x_k(b)] &= \frac{1}{2} \left[ \frac{1}{x_i(a) + x_i(b)} + \frac{1}{x_k(a) + x_k(b)} \right] \\ &\times \left[ \frac{1}{(x_i(a) + x_k(a))(x_i(b) + x_k(b))} \right. \\ &\left. + \frac{1}{(x_i(a) + x_k(b))(x_i(b) + x_k(a))} \right] \end{aligned} \quad (69)$$

と定義されます。さらに各変数は

$$x_j(a) = \frac{a^2}{2} \times \frac{1}{1 - \exp(-4\pi a^2 / 9d_j^2)}, \quad (70)$$

$$d_j = |\mathbf{r}_i - \mathbf{r}_k| q_0(\mathbf{r}_j), \quad (71)$$

$$q_0(\mathbf{r}_j) = -\frac{4\pi}{3} \epsilon_{xc}^{\text{LDA}} \rho(\mathbf{r}_j) - \frac{Z_{ab}}{9} \left\{ \frac{\nabla \rho(\mathbf{r}_j)}{2k_F(\mathbf{r}_j) \rho(\mathbf{r}_j)} \right\}^2 k_F(\mathbf{r}_j), \quad (72)$$

$$k_F^3(\mathbf{r}_j) = 3\pi^2 \rho(\mathbf{r}_j) \quad (j = i \text{ or } k) \quad (73)$$

となっており、これからわかるように電荷密度分布を唯一の入力情報とした汎関数となるように設計されています．ここで式 (72) 内の定数  $Z_{ab} = -0.8491$  は第一原理的に決定された係数です．局所密度近似による交換相関エネルギー密度  $\epsilon_{xc}^{LDA}$  は O. Gunnarsson et al [14] によるものを用いています．これら一連の式は plasmon-pole model をもとに設計されたものであり、そのため van der Waals 相互作用に代表される非局所相関項を含む全エネルギーが比較的計算コストで非経験的に得られるようになっています．

さらに効率的に数値計算を行うために、式 (66) の 2 変数関数  $\phi(\mathbf{r}_i, \mathbf{r}_k)$  を計算するアルゴリズムが変更されました．この 2 変数関数  $\phi(\mathbf{r}_i, \mathbf{r}_k)$  は直接には di と dj にしか依存しないため、 $d_i = D(1 + \dots)$ ,  $d_j = D(1 - \dots)$  と新たに定義された 2 変数  $D$ ,  $\delta$  を用いて  $\phi(\mathbf{r}_i, \mathbf{r}_k)$  を  $\phi(D, \delta)$  として予め計算された数値セットを用意するようにしました．これによって用意したグリッド点数に応じて式 67 の 2 重積分を逐一行う必要がなくなるため数値計算量が大幅に削減されています．

特異点周辺の数値積分 式 (70) で変数  $a = 0$  でかつ  $\mathbf{r}_i = \mathbf{r}_k$  の場合は数値計算によって  $x_j(a)$  を決定するのは困難です．このため式 (66) の数値積分も特異点を含むことになるため、難しくなります．そこで  $|\mathbf{r}_i - \mathbf{r}_k| \ll 1$  の領域では電荷密度を  $\rho(\mathbf{r}_i) = \rho(\mathbf{r}_k)$  と仮定して  $\mathbf{r}_k$  積分の外に出すことにします．これによって式 (66) のうち、 $\mathbf{r}_i$  を中心にした微小半径  $\eta_i$  の球内での  $\mathbf{r}_k$  積分を

$$\begin{aligned} \frac{1}{2} \int_{\eta_i} d\mathbf{r}_k \rho(\mathbf{r}_i) \phi(\mathbf{r}_i, \mathbf{r}_k) \rho(\mathbf{r}_k) &\cong \frac{1}{2} \frac{4\pi\eta_i^3}{3} \rho^2(\mathbf{r}_i) \int_{\eta_i} d\mathbf{r}_k \phi(\mathbf{r}_i, \mathbf{r}_k) \\ &= \frac{1}{2} \frac{4\pi\eta_i^3}{3} \rho^2(\mathbf{r}_i) \times 4\pi \int_0^{\eta_i} dr_{ik} \phi(d_i, d_k) \\ &= \frac{1}{2} \frac{4\pi\eta_i^3}{3} \rho^2(\mathbf{r}_i) \times 4\pi \int_0^{\eta_i q_0} dD \phi(D) \frac{D^2}{q_0^3} \end{aligned} \quad (74)$$

と単純にすることができます．ただし、 $r_{ik} = |\mathbf{r}_i - \mathbf{r}_k|$  としています．ここで 2 行目から 3 行目へは、 $D \equiv q_0 r_{ik} (= d_i = d_k)$  を定義して変数変換を用いています．3 行目の被積分関数は特異点を含まないため、これで数値積分を実行することができます．

無限周期系への拡張 本手法では電荷密度分布情報を実空間表記で入力し、そのまま計算を進めていくためにこのままでは有限サイズの計算対象に向けた手法となっています．逆に無限周期系を計算するには十分に  $\mathbf{r}_i$  と  $\mathbf{r}_k$  が離れた場合まで式 (66) の数値積分を行わなくてはならず、非常に計算効率が悪くなります．特に van der Waals 相互作用は遠距離においてもその寄与はなだらかにしか減衰しません．そこで漸近関数を用いることで、本手法をこの無限周期系にも対応できるようにしました．式 (66) の被積分関数にある 2 変数関数  $\phi(\mathbf{r}_i, \mathbf{r}_k)$  は  $\mathbf{r}_i$  と  $\mathbf{r}_k$  が十分に離れた場合には

$$\begin{aligned} \phi(\mathbf{r}_i, \mathbf{r}_k) &\rightarrow -\frac{12(4\pi/9)^3}{d_i^2 d_k^2 (d_i^2 + d_k^2)} \\ &= -\frac{C}{r_{ik}^6} \times \frac{1}{q_0^2(r_i) q_0^2(r_k) (q_0^2(r_i) + q_0^2(r_k))} \\ &= -\frac{C}{r_{ik}^6} \times \psi[\rho(\mathbf{r}_i) \rho(\mathbf{r}_k)] \end{aligned} \quad (75)$$

のような漸近関数に近似できることがその定義からわかります．ただし  $C = 12(4\pi/9)^3$ ．この漸近関数には式 (67) のような積分演算が含まれていないため非常に簡単に数値計算が行えるという利点があります．この式の 3 行目のうち、 $\times$  の前の分数は増加に応じて単純に減衰していく漸近項で、これに対して  $\times$  の後ろの  $\psi[\rho(\mathbf{r}_i) \rho(\mathbf{r}_k)]$  は式 (72) からわかるように直接的には電荷密度しか変数を持たないため、周期系物質においては周期的な項となっています．このことを考慮すると、式 (66) のうち、 $\mathbf{r}_i$  と  $\mathbf{r}_k$  が十分に離れた場合 ( $r_{ik} > \eta$ ) には次のように積分を単純化できます．すなわち

$$\begin{aligned} \frac{1}{2} \int_{r_{ik} > \eta} d\mathbf{r}_i d\mathbf{r}_k \rho(\mathbf{r}_i) \phi(\mathbf{r}_i, \mathbf{r}_k) \rho(\mathbf{r}_k) &\cong \frac{1}{2} dv^2 \sum_{\substack{\mathbf{r}_i \in \\ \text{unitcell}}} \sum_{\substack{\mathbf{r}_k \in \\ (r_{ik} > \eta)}} \rho(\mathbf{r}_i) \phi(\mathbf{r}_i, \mathbf{r}_k) \rho(\mathbf{r}_k) \\ &= \frac{1}{2} dv^2 \sum_{\substack{\mathbf{r}_i, \mathbf{r}_k \in \\ \text{unitcell}}} \sum_{\substack{\mathbf{t}_{xyz} \in \\ (|\mathbf{r}_k + \mathbf{t}_{xyz} - \mathbf{r}_i| > \eta)}} \rho(\mathbf{r}_i) \phi(\mathbf{r}_i, \mathbf{r}_k + \mathbf{t}_{xyz}) \rho(\mathbf{r}_k + \mathbf{t}_{xyz}) \end{aligned} \quad (76)$$

$$= -\frac{C}{2} dv^2 \sum_{\substack{\mathbf{r}_i, \mathbf{r}_k \in \\ \text{unitcell}}} \rho(\mathbf{r}_i) \psi[\rho(\mathbf{r}_i), \rho(\mathbf{r}_k)] \rho(\mathbf{r}_k) \sum_{\substack{\mathbf{t}_{xyz} \in \\ (|\mathbf{r}_k + \mathbf{t}_{xyz} - \mathbf{r}_i| > \eta)}} \frac{1}{r_{xyz}^6}$$

と関数を分けることができます．ここで  $dv$  は数値計算を行う際のグリッド点に囲まれる最小直方体体積で， $\mathbf{t}_{xyz}$  は格子点同士を結ぶ結晶ベクトルです．この式 3 行目の総和のうち左側は  $\mathbf{r}_i$  と  $\mathbf{r}_k$  の範囲がユニットセル内に限定されているので，計算コストは限定的です．一方，右側は  $1/r_{xyz}^6$  が実質無視できるほど小さくなるまで広範囲に渡って総和することになります．しかし，そもそも関数型が非常に単純な上に，電荷密度分布  $\rho(\mathbf{r})$  とは無関係で，グリッド点の空間配置のみによって決まる値なので予め求めておけばよいものです．以上のように漸近関数を使ってさらに式 (76) のように積分順序を工夫することによって，実質無限遠の寄与まで効率よく考慮することができ，これによって本手法を無限周期系にも適用できるようにしています．

アルゴリズム vdW-DF 計算を非セルフコンシステント（いわゆる 1 shot 計算）に求めるプログラムルーチンが整備されています．このルーチンは通常の GGA から相関項を除外した計算をセルフコンシステントに行った際の出力を用いて，非局所相関項  $E_c^{nl}$  および局所相関項  $E_c^{LDA}$  を 1 shot 計算により決定します．最後にこれら 2 つの相関項を，上記の相関項を除外した GGA 計算（すなわち交換項のみ）から得られた全エネルギー  $E_{total}^{GGAx}$  に加えることで「非局所相関項も考慮した全エネルギー  $E_{total}^{vdW-DF}$ 」を計算します．図 38 は本ルーチン（vdW）を用いた計算の流れを示しています．緑枠が vdW で，青枠が交換項のみの GGA ルーチンです．つまり，交換項のみの GGA 計算をセルフコンシステントに行い，電荷密度  $\rho^{GGAx}(\mathbf{r})$  と  $E_{total}^{GGAx}$  を出力させます．続いて vdW に  $\rho^{GGAx}(\mathbf{r})$  を入力させ， $E_c^{nl}$  および  $E_c^{LDA}$  を 1 shot で計算させます．最後にこれら 3 つのエネルギー項を足し合わせることで  $E_{total}^{vdW-DF}$  とします．

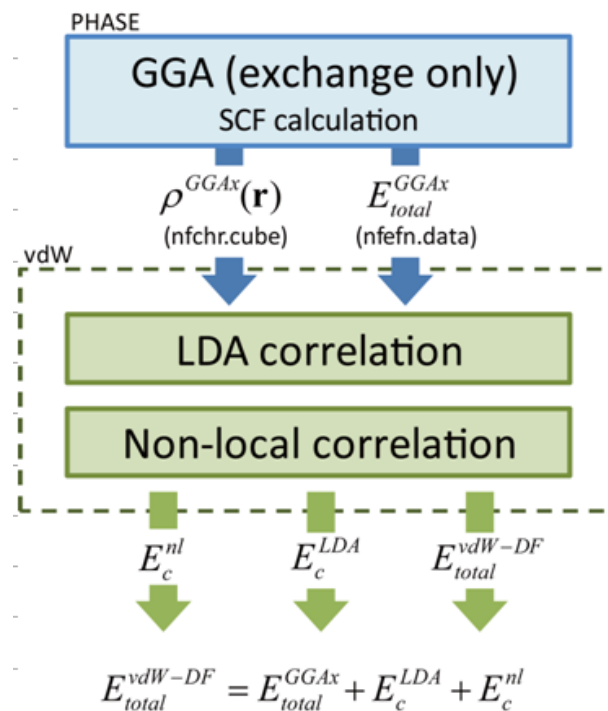


図 38: 本ルーチン（vdW）の計算の流れ

### 19.2.2 入出力の説明

前段階としての PHASE 本ルーチン（vdW）は `vdW.f90` というファイル内で全て完結しています．これはポスト処理用に設計されており，PHASE の交換項のみを考慮した GGA 計算から得られる電荷密度分布  $\rho^{GGAx}(\mathbf{r})$  の数値ファイル `nfchr.cube` と全エネルギー値ファイル `nfefn.data` を入力し，相関エネルギー  $E_c^{nl}$  と  $E_c^{LDA}$  を出力します．最後に GGA 計算（交換項のみ考慮）による全エネルギーにこれらを加えた値  $E_{total}^{vdW-DF}$  を出力します．

(図 38 参照) このため GGA 計算時に `nfchr.cube` を出力するように PHASE の `inputfile` を設定する必要があります。また、擬ポテンシャルは PBE のものを前提としています。これらを踏まえて PHASE の `file_names.data` と `inputfile` 内で `vdW` を実行するためには以下のような設定が必要になります。

```
file_names.data (PHASE) :
    F_ENF    = ' ./nfefn.data '
    F_CHR    = ' ./nfchr.cube '

inputfile (PHASE) :
accuracy{
    xctype = ggapbex
}

postprocessing{
    charge{
        sw_charge_rspace = ON
        filetype         = cube
    }
}
```

PHASE 実行時には `ggapbe` 型の擬ポテンシャルを用いるのですが、この際相関項を無視する必要があります。そこで、これまでの PHASE の擬ポテンシャル識別子「`xctype`」用に新たに「`ggapbex`」という項目を追加しました。「`xctype = ggapbex`」とすることで `ggapbe` のうち交換項のみを考慮するように指定できます。指定する擬ポテンシャルは `ggapbe` の場合と同じにします。

**vdW.f90 のコンパイル** 本プログラム (`vdW.f90`) は Fortran90 コンパイラによってコンパイルされます。また、OpenMP によって並列化されているため、並列計算用にもコンパイルができます。Intel compiler の場合は単に `-openmp` のオプションを加えるだけで並列計算ができます。コンパイル例：

```
$ ifort -openmp -o vdW vdW.f90
```

**ポスト処理としての vdW** 続いて `vdW` の入力について、`vdW` は、PHASE で得られる `nfchr.cube` と `nfefn.data` が同ディレクトリ内に存在していれば自動的にそれらを入力します。このため `vdW` 専用の入力ファイルを新たに作成する必要はありません。ただしファイル名は `nfchr.cube`、`nfefn.data` とし、変更できません。

**出力について** `vdW` を実行した結果は以下のようなフォーマットで出力されます。単位は通常の PHASE と同じく Hartree です。

出力例：(例として `phase/samples/vdW/input_scf.Si.data` からの出力ファイル `nfchr.cube` と `nfefn.data` を用いて計算しています。)

```
E_total(GGA exchange) = -7.5363221703000

Ec(LDA) = -0.5429739815997
Ec(nl) = 0.0203272639208
Ec (= Ec(LDA) + Ec(nl) ) = -0.5226467176789

E_total(vdW-DF) = -8.0589688879789

Given in Hartree atomic units

# Calculation time 0 : 11 : 33.7280
```

各変数の説明：

E <sub>total</sub> (GGA exchange)	GGA (exchange) の全エネルギー
Ec(LDA)	LDA による局所的相関項
Ec(=Ec(LDA)+Ec(nl))	全相関項
E <sub>total</sub> (vdW-DF)	vdW-DF 計算を含めた全エネルギー
Calculation time	時：分：秒

E<sub>total</sub>(vdW-DF) がこの vdW の主目的である「非局所相関項を含むエネルギー」です。

計算例 以上を実際に再現するために、phase/samples/vdW/内に例を用意しました。vdW/内の file\_names.data 及び input\_scf.Si.data を用いてまずは phase を実行し、nfchr.cube と nfeffn.data を出力させます。同ディレクトリ内に vdW.f90 が用意してあるのでこれをコンパイルし、そのまま実行をします。

計算手順例：

```
$ cd phase/sample/vdW/ (例の用意されているディレクトリに移動)
$ ../../bin/phase (phase による GGA 計算の実行)
$ ifort -openmp -o vdW vdW.f90 (同ディレクトリ内にある vdW.f90 をコンパイル)
$ ./vdW (そのまま実行)
```

出力結果が、上の出力例のようになるはずです。

### 19.2.3 積層グラファイトの全エネルギー計算

はじめに ここでは実際に本ルーチン(vdW)を用いた vdW-DF 計算の例を挙げます。計算対象は GGA や LDA では正確に計算できない典型例である、積層グラファイト(A-B stacking)の全エネルギーの層間距離依存性としてしました。通常の GGA の範囲内でこの系の全エネルギー曲線を計算するとエネルギー的に安定な平衡点は現れず、結果的に各グラファイト層は互いに無限遠まで離散するという解釈になってしまいます。本来は適当な層間距離でエネルギー的に明確な安定点が存在するため、GGA によるこの解釈は定性的に間違っています。これは、非局所的な相互作用である van der Waals 相互作用を GGA が全く考慮できていないことに主な要因があり、本ルーチンを利用することでこの間違いが修正されることを確認します。

計算条件 本計算では、実験値や他の理論計算の報告例が豊富にある A-B stacking 型の積層グラファイトを対象にしました。現段階の本ルーチンの制約上、直方体型のユニットセルを設定し、それぞれに 8 個(2 層分)の炭素原子を含みます。グラファイトの積層方向に z 軸をとってセルサイズは  $4.3 \times 2.5 \times z$  ( $=x \times y \times z [\text{\AA}^3]$ ) とし、z を 5 から 12[Å] まで変化させながら各 z 値での全エネルギーを計算させました。この計算を通常の GGA と vdW-DF 計算の 2 通りで行い、それぞれから得られる全エネルギー曲線を比較します。vdW-DF 計算には入力情報として電荷密度分布  $\rho^{\text{GGAx}}(\mathbf{r})$  が必要になりますが、このグリッド密度は GGA 計算時に設定した cutoff 値に依存しており、ここでは  $32 \times 18 \times 40$ (to 96) 個のグリッドを採用しています。なお、グリッド点数は整数であるため、z 軸の変化に伴って不連続に変化することになります。つまり、結果のエネルギー曲線が不自然に変化する場合はこの不連続性が原因であるため、必要に応じて PHASE の inputfile の cutoff 値を上げることでなめらかになります。また、vdW は非局所的な計算になるため GGA よりは大きな計算コストが必要となります。

計算結果 図 39 は上記条件で計算した積層グラファイト(A-B stacking)の全エネルギーの層間距離依存性を示したものです。赤線は通常の GGA による結果を、緑線は vdW による vdW-DF 計算の結果をそれぞれ示しています。また、青点は実験 [15,16] による平衡点とそのときの凝集エネルギー(青線は誤差)を、同様に紫点は別の理論計算 [17] によるものです。GGA 計算では全く平衡点が確認できないのに対して、vdW-DF 計算ではかなり実験値の近くで極小点を迎えているのが確認できます。

## 19.3 非局所相関項を含めた電子状態計算(セルフ・コンシステント計算用)

### 19.3.1 理論概要

Van der Waals 相互作用を加えた全エネルギーを導出するためには先の節で記述したポスト計算用のプログラム  $\Delta$  vdW.F90 を用いればよいですが、バンド図や電荷密度分布などの電子状態や原子にかかる力などを導出するためには Kohn-Sham 方程式に直接挿入してセルフ・コンシステントに計算する必要があります。なお厳密には全エネルギーもセルフ・コンシステント計算によって求める必要がありますが、vdW 項による電荷密度の変化は量的には僅かであり、経験的に計算コストの軽いポスト計算でもほとんど差し支えないことが確認されています [18]。

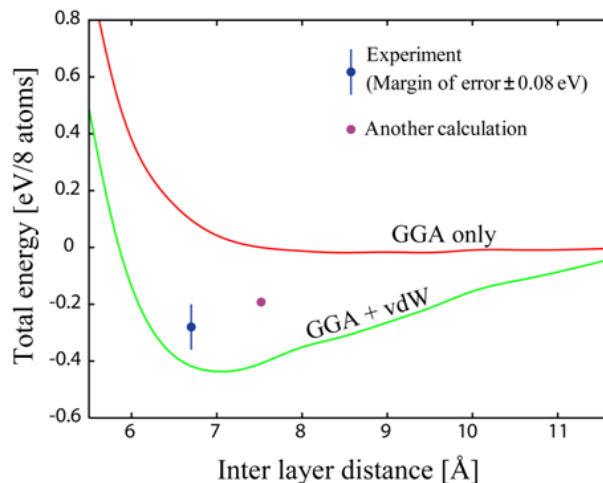


図 39: 積層グラファイトの全エネルギーの層間距離依存性．GGA（赤線）と GGA + vdW（緑線）による比較

DFT で用いる Kohn-Sham 方程式は変分法を用いて解かれるため vdW 項を加える際にもエネルギー項を密度によって汎関数微分した非局所相関ポテンシャルを用意する必要があります．つまり

$$\begin{aligned} v_c^{\text{nl}} &= \delta E_c^{\text{nl}} / \delta \rho(\mathbf{r}) \\ v_c^{\text{LDA}} &= \delta E_c^{\text{LDA}} / \delta \rho(\mathbf{r}) \end{aligned} \quad (77)$$

のようにエネルギー項を電荷密度で汎関数微分した後に，

$$v_{\text{xc}} = v_{\text{x}}^{\text{GGA}} + v_c^{\text{LDA}} + v_c^{\text{nl}} \quad (78)$$

を Kohn-Sham 方程式内に直接挿入します．式 (67) からわかるように， $E_c^{\text{nl}}$  や  $E_c^{\text{LDA}}$  は電荷密度  $\rho(\mathbf{r})$  を簡単なかたちで含むために汎関数微分を解析的に求めることができます．つまり，

$$v_c^{\text{nl}} = \int d^3\mathbf{r}_k \rho(\mathbf{r}_k) \sum_{j=1}^4 \alpha_j(\mathbf{r}_i, \mathbf{r}_k) \Phi(\mathbf{r}_i, \mathbf{r}_k) \quad (79)$$

となります．[18] ここで  $\alpha_j$  や  $\Phi_j (j = 1 - 4)$  は式 (66) の汎関数  $\phi$  を汎関数微分する際に発生する項です．

### 19.3.2 入出力の説明

本ルーチンは GGA の交換ポテンシャル  $v_{\text{x}}^{\text{GGA}}$  に局所相関ポテンシャル  $v_c^{\text{LDA}}$  および非局所相関ポテンシャル  $v_c^{\text{nl}}$  を加えた Kohn-Sham 方程式をセルフ・コンシステントに計算します．PHASE で GGA の計算と同様に実行ファイル phase を実行するだけで，vdW 相互作用を考慮した電子状態が計算できます．また，原子にかかる力も vdW 相互作用が考慮されているものが出力ファイル nfdynm.data に従来と同様に出力されます．PHASE の入力ファイル内で vdW 項を含めたセルフ・コンシステント計算に必要な新たな設定は以下のようになります．

```
inputfile (PHASE) :
accuracy{
  xctype = vdwdf
}
```

このセルフ・コンシステント計算用のプログラム vc\_nl.F90 は PHASE のディレクトリ src\_phase/内に置いてあります．src\_phase/で従来の PHASE 同様コンパイルが出来ます．また vc\_nl.F90 内は OpenMP による並列化がなされています．

## 20 ハイブリッド汎関数法

### 20.1 概説

厳密交換エネルギーは  $\sigma$  スピンの  $n$  番目の状態の波動関数  $\psi_{n\sigma}(\mathbf{r})$  を用いて

$$E_x^{\text{exact}} = -\frac{1}{2} \sum_{\sigma} \sum_{n,m}^{\text{occ}} \int d\mathbf{r}_1 \int d\mathbf{r}_2 \frac{\psi_{n\sigma}^*(\mathbf{r}_1) \psi_{m\sigma}(\mathbf{r}_1) \psi_{m\sigma}^*(\mathbf{r}_2) \psi_{n\sigma}(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} \quad (80)$$

で与えられる。 $n, m$  に関する和は占有状態に限る。

厳密交換エネルギーを PBE 汎関数に混合させたハイブリッド交換相関汎関数は

$$E_{\alpha}^{\text{hybrid}} = \alpha E_x^{\text{exact}} + (1 - \alpha) E_x^{\text{PBE}} + E_c^{\text{PBE}} \quad (81)$$

と表せる。ここで、 $E_x^{\text{PBE}}$  は PBE 交換汎関数であり、 $E_c^{\text{PBE}}$  は PBE 相関汎関数である。混合比率  $\alpha = \frac{1}{4}$  の時、 $E_{\alpha}^{\text{hybrid}}$  は PBE0 汎関数になる。

短距離交換相互作用  $V_{sr} = \text{erfc}(\omega r)/r$  に基づく短距離厳密交換エネルギー  $E_x^{\text{exact,sr}}$  を PBE 汎関数に混合させた HSE ハイブリッド汎関数は

$$E_{\alpha}^{\text{HSE}} = \alpha (E_x^{\text{exact,sr}} - E_x^{\text{PBE,sr}}) + E_x^{\text{PBE}} + E_c^{\text{PBE}} \quad (82)$$

と表せる。ここで、 $E_x^{\text{PBE,sr}}$  は PBE 汎関数の短距離項である。混合比率  $\alpha = \frac{1}{4}$  の時、遮蔽パラメータ  $\omega$  を 0.106 とすると HSE06 汎関数になる。

### 20.2 使用方法

HSE06 ハイブリッド汎関数で電子状態計算を行うには、以下の設定を使います。

```
accuracy{
  xctype = ggapbe
  hybrid_functional{
    sw_hybrid_functional = ON
    sw_screened_exchange = ON
    alpha = 0.25
    omega = 0.106
  }
}
```

PBE0 ハイブリッド汎関数で電子状態計算を行うには、sw\_screened\_exchange を OFF にしてください。分子の計算などで  $\Gamma$  点のみをサンプリングする場合は、以下のような通常の設定が使用できます。

```
accuracy{
  ksampling{
    method = gamma
  }
}
```

バンド計算では、以下のような  $k$  点サンプリングの設定を使います。

```
accuracy{
  ksampling{
    method = monk
    mesh{ nx = 6, ny = 6, nz = 6 }
    kshift{ k1=0, k2=0, k3=0 }
    base_reduction_for_GAMMA = OFF
  }
}
```



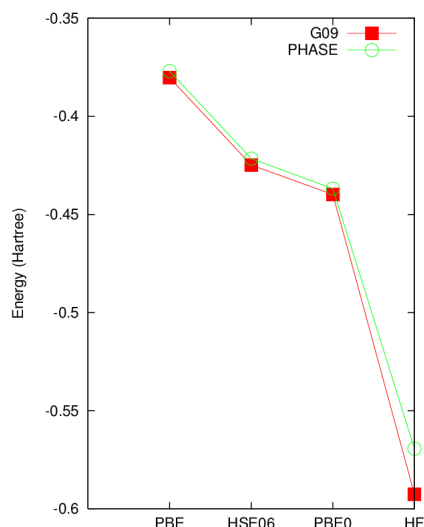


図 40: PBE 汎関数法,HSE06 汎関数法,PBE0 汎関数法,HF 法による水素分子の HOMO 準位。

$nx, ny, nz$  は収束性を確認して適切な値を使用してください。さらに、以下のように入力して、PBE 汎関数で収束させた波動関数と電荷密度をハイブリッド汎関数計算の初期波動関数と初期電荷としてください。

```
accuracy{
    initial_wavefunctions = file
    initial_charge_density = file
}
```

PBE 汎関数計算の波動関数ファイル (zaj.data) と電荷密度ファイル (nfchgt.data) を作業ディレクトリーに必ずコピーしてください。使用可能なソルバーは SD 法、MSD 法、lmMSD 法、RMM3 法に限られていますので、ソルバー設定には注意を要します。サンプルのソルバー設定を参考に設定してください。擬ポテンシャルはノルム保存型のみ使用できます。

また、以下の入力を用いると、Hartree-Fock 計算ができます。

```
accuracy{
    hybrid_functional{
        sw_hybrid_functional = ON
        sw_exchange_only = ON
        alpha = 1.0
    }
}
```

## 20.3 計算例

### 20.3.1 水素分子

サンプルディレクトリ samples/hybrid/H2 の下のディレクトリ PBE,HSE06,PBE0,HF に PBE 計算、HSE06 計算、PBE0 計算、HF 計算の入力サンプルがあります。go\_h2.sh を実行すると、これらの計算を順番に実行します。これらの結果を図 40 に示します。

### 20.3.2 Si 結晶

サンプルディレクトリ samples/hybrid/Si の下のディレクトリ PBE,HSE06 に PBE 計算、HSE06 計算の入力サンプルがあります。go\_si.sh を実行すると、これらの計算を順番に実行します。HSE06 汎関数で計算した結果を図 41 に示します。ただし、この結果は以下のように計算条件を変更して得たものです。

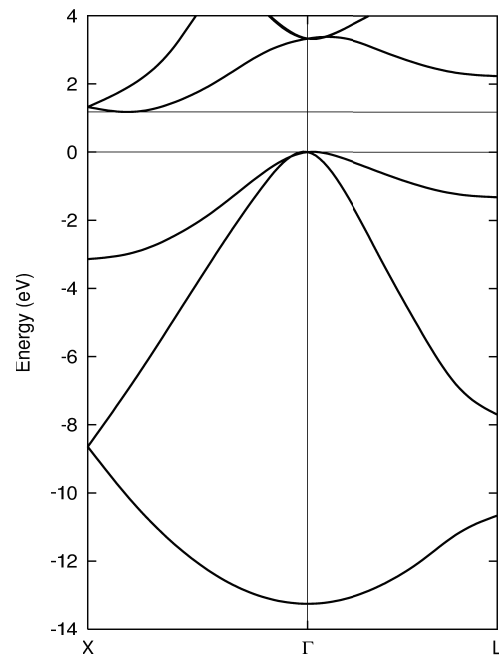


図 41: HSE06 汎関数法による Si 結晶のバンド構造。

```

accuracy{
  cutoff_wf = 20.00 rydberg
  cutoff_cd = 80.00 rydberg
  ksampling{
    method = monk
    mesh{ nx = 10, ny = 10, nz = 10 }
    kshift{ k1=0, k2=0, k3=0}
    base_reduction_for_GAMMA = OFF
  }
}

```

## 21 PAW 法

### 21.1 利用方法

PAW 法を利用するためには、PAW ポテンシャルをダウンロードしていただく必要があります。http://www.ciss.iis.u-tokyo.ac.jp/dl/index.php よりダウンロードし、解凍してください。利用するには、通常のノルム保存およびウルトラソフトポテンシャルと同様、file\_names.data ファイルにおいて F\_POT 識別子で対応する元素名のファイルを指定します。

ekcal による固定電荷の計算を行う場合、さらに file\_names.data に F\_CNTN\_BIN\_PAW 識別子を利用して PAW 計算用のデータを指定し、読み込ませる必要があります。このファイルの既定のファイル名は continue\_bin\_paw.data です。たとえば SCF 計算をディレクトリーを行ったディレクトリーが 1 階層上のディレクトリーだった場合、下記のような記述が必要です。

```
&fnames
...
...
F_CNTN_BIN_PAW='../continue_bin_paw.data'
/
```

### 21.2 計算例

PAW 法を利用した計算例として、体心立方構造クロムの格子定数の計算例を紹介します。クロムは、ウルトラソフト擬ポテンシャルで計算すると格子定数が過大評価され、また体積弾性率が過小評価されます。この点がどのように改善されるか確認します。

計算に利用したデータは、samples/Cr 以下にあります。samples/Cr 以下には、以下のファイルがあります。

```
Cr_ggapbe_paw_002.gncpp2
paw/
us/
```

Cr\_ggapbe\_paw\_002.gncpp2 が PAW 用のポテンシャルファイルです。PAW 用の擬ポテンシャルファイルは、入力に以下を記述することによってウルトラソフト型として利用することも可能です。

```
accuracy{
    paw_switch = off
}
```

ディレクトリー paw に PAW 用の入力データが、us にウルトラソフト用の入力データが格納されています。paw, us の各ディレクトリーには下記のディレクトリーが存在します。

```
catenergy.sh
vol20/
vol21/
...
```

catenergy.sh は、すべての体積についての計算が終了したあとにエネルギーデータファイルを結合するシェルスクリプトです。vol20, vol21, ... はそれぞれ体積  $20\text{\AA}^3$ ,  $21\text{\AA}^3$ , ... の入力データに対応します。すべてのディレクトリーで計算を実行したあと、catenergy.sh を実行すると paw(us) ディレクトリーに各体積のエネルギーを記録したファイル、energy.data が作成されます。

paw ディレクトリー下の入力データが格納されたディレクトリーの file\_names.data ファイルには、以下のような記述があります。

```
F_POT(1) = '../..../Cr_ggapbe_paw_002.gncpp2'
F_POT(2) = '../..../Cr_ggapbe_paw_002.gncpp2'
```

この指定があるので、PAW 用のポテンシャルが利用されます。ディレクトリー us の入力ファイルには、上記の paw\_switch = off の指定があるので、ウルトラソフト型の計算が実行されます。

計算の結果得られた  $EV$  曲線を図 42 に示します。一見して明らかなように、PAW 法とウルトラソフト法とでは全くことなる  $EV$  曲線が得られます。さらに、この  $EV$  曲線をもとに格子定数と体積弾性率をもとめた結果を実測値とともに表 13 にまとめました。US 法と比較して PAW 法によって実測値との一致が改善していることが分かります。

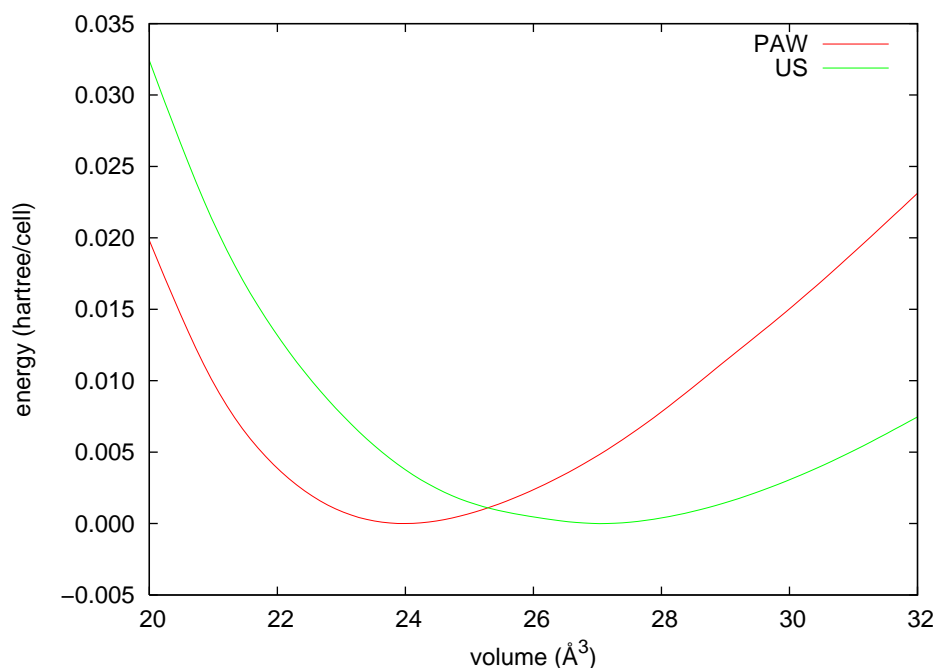


図 42: クロムの  $EV$  曲線。赤線が PAW の結果，緑線が US の結果。各手法で得られた最も低いエネルギーをエネルギーの原点としている。

表 13: PAW 法および US 法によってもとめた格子定数と体積弾性率

	US	PAW	実測
$a$ ( $\text{\AA}$ )	2.994	2.886	2.88
$B$ (GPa)	89.2	150.5	190.1

## 22 Q and A

ここで Q and A をまとめてみます。

1.
  - Q : ekcal を使って計算したバンド分散の結果がおかしいのですが？
  - A : 計算された波動関数は十分に収束していますか？ ログファイル output000 を調べてみましょう。
2.
  - Q : SCF 計算の途中で、電荷密度が負になるというエラーがでるのですが？  
(例：\*\* WARN CHG.DEN < 0.0 AT 1 - 0.8220751D-1)
  - A : charge mesh の精度は充分でしょうか？電荷密度のカットオフを指定する cutoff.cd の値を大きくして計算してみましょう。  
(なお、SCF 計算の初期の段階にのみ、上記 warning が現れる場合は、問題ありません。)
3.
  - Q : cutoff\_wf や cutoff\_cd の値はどうやって決めればよいのですか？
  - A : cutoff\_wf の値は用いる擬ポテンシャルに依存します。通常は、結晶の計算をし、格子定数や、体積弾性率の実験値を再現する様に値を決めます。通常の TM ポテンシャルの場合は cutoff\_cd = 4 × cutoff\_wf となります。ただし、擬ポテンシャルが PCC を使っている場合やウルトラソフトポテンシャルの場合は、一般にそれよりも大きくする必要があります。
4.
  - Q : mpirun が動作しません。
  - A : .rhost にマシン名とユーザーアカウント名とを書いていますか。
5.
  - Q : 電子状態の最適化がうまくいきません (電荷密度が収束しません)。
  - A : 第 8 節を参考にして、最適化のオプションを変えて試して下さい。
6.
  - Q : 入力ファイルで xctype を指定しますが、擬ポテンシャルデータ中にある xctype と一致している必要がありますか？
  - A : 合理的な計算をするために、原則的に必要です。入力で xctype を指定しなかった場合、擬ポテンシャルのファイル中に書かれている xctype の値が使われます。尚、複数の擬ポテンシャルが使われる場合、各擬ポテンシャルデータ中に書かれている xctype の種類が一致していないと、job は実行されません。
7.
  - Q : 金属を対象に計算をして band.pl を使ってバンド構造図を描画したら、バンド構造図のフェルミレベルがずれてしまうのですが、どうしたらよいのでしょうか？
  - A : nfenenergy.data のフェルミエネルギーの値を、scf のフェルミエネルギーの値と同じにしてください。
8.
  - Q : band.pl (あるいは dos.pl) が使えません。
  - A : Perl スクリプトは実行可能になっていますか。chmod +x dos.pl 等の処理をしてから、もう一度試してみてください。これらの Perl スクリプトの使用方法については、tools のマニュアルをご参照下さい。
9.
  - Q : SCF の収束の条件を満たしていないように見えるのに、計算が終了するのはなぜですか。
  - A : 収束条件は、1 原子当たりのエネルギーに対して設定されています。これに対して、PHASE の標準出力結果である output000 に出力されているのは全エネルギーなので、これを原子数で割ったものが、収束条件を満たしているかどうかを確認しなければなりません。
10.
  - Q : 計算結果が本当に収束したかどうかは、出力ファイルのどこを見れば確認できるのでしょうか。
  - A : SCF 計算の結果が収束していれば、出力ファイルである nfeffn.data や nfdynm.data に何か書き込まれているはずです。原子に働く力の計算が収束したならば、continue.data の最後に、
 

```
convergence
      2
```

 という記述が見つかるはずです。
11.
  - Q : チュートリアルを図 3 にある青線のユニットセルは、どの様にすれば描画できますか？
  - A : 以下の mol2 file を作成し、biostation viewer で読み込んでください。

```

@<TRIPOS>MOLECULE
Si8 frame
      8    12    0
      0    0    0    0

grid file
@<TRIPOS>ATOM
      1 N      0.000000  0.000000  0.000000   N.4   1  GLY      0.0000
      2 N      5.430000  0.000000  0.000000   N.4   1  GLY      0.0000
      3 N      0.000000  5.430000  0.000000   N.4   1  GLY      0.0000
      4 N      0.000000  0.000000  5.430000   N.4   1  GLY      0.0000
      5 N      0.000000  5.430000  5.430000   N.4   1  GLY      0.0000
      6 N      5.430000  0.000000  5.430000   N.4   1  GLY      0.0000
      7 N      5.430000  5.430000  0.000000   N.4   1  GLY      0.0000
      8 N      5.430000  5.430000  5.430000   N.4   1  GLY      0.0000
@<TRIPOS>BOND
      1      1  2      1
      2      1  4      1
      3      1  3      1
      4      2  6      1
      5      2  7      1
      6      3  5      1
      7      3  7      1
      8      4  5      1
      9      4  6      1
     10      5  8      1
     11      6  8      1
     12      7  8      1

```

同様の描画は、PHASE-Viewer を利用しても実行することができます。詳しくは PHASE-Viewer マニュアル 8.10 節「単位胞」をご覧ください。

## 23 終りに

第一原理密度汎関数法による電子状態の計算プログラム PHASE の使い方について説明しました。目的に合わせて入力ファイルを書き換えることにより、機能をいろいろ組み合わせた計算が可能なが、御理解頂けましたでしょうか。今後の皆様の研究や開発に、PHASE をお役立て下さることを願っております。

## 参考文献

- [1] H. Kageshima, “Theoretical Simulation of Scanning Tunneling Microscopy/Spectroscopy -Beyond LCAO Approximation”, Doctor Thesis of University of Tokyo (1990).
- [2] 加藤氏のノートと聞き書きによる
- [3] G. Mills and H. Jónsson, “Quantum and Thermal Effects in H<sub>2</sub> Dissociative Adsorption: Evaluation of Free Energy Barriers in Multidimensional Quantum Systems” *Phys. Rev. Lett.* **72** (1994) p. 1124.
- [4] G. Henkelman, B. P. Uberuaga and H. Jónsson, “A climbing image nudged elastic band method for finding saddle points and minimum energy paths” *J. Chem. Phys.* **113** (2000) p. 9901.
- [5] M. J. Puska, A. P. Seitsonen, and R. M. Nieminen, “Electron-positron Car-Parrinello Methods: Self-consistent Treatment of Charge Densities and Ionic Relaxations”, *Phys. Rev. B* **52** (1995) p. 10947.
- [6] M. J. Puska, “Ab-initio Calculations of Positron Annihilation Rates in Solids”, *J. Phys. Condens. Matter* **3** (1991) p. 3455.
- [7] A. Nakamoto, M. Saito, T. Yamasaki, M. Okamoto, T. Hamada, and T. Ohno, “Two-Component Density Functional Calculations on Positron Lifetimes for Band-Gap Crystals”, *Jpn. J. Appl. Phys.* **47** (2008) p. 2213.
- [8] G. Pelz, K. Yamada, and G. Winnewisser, *Journal of Molecular Spectroscopy* **159**, (1993) p. 507.
- [9] Michiel Sprik and Giovanni Ciccotti, *Journal of Chemical Physics* **109** (1998) p. 7737.
- [10] A. Laio and M. Parrinello, *Proceedings of the National Academy of Sciences* **99**, (2002) p. 12562.
- [11] M. Iannuzzi, A. Laio and M. Parrinello, *Physical Review Letters* **90**, (2003) p. 238302.
- [12] A. Laio, A. Rodriguez-Forteza, F. L. Gervasio, Ceccarelli and M. Parrinello, *J. Phys. Chem. B* **109**, (2005) p. 6714.
- [13] M. Dion, H. Rydberg, E. Schroder, D. C. Langreth, and B. I. Lundqvist: *Phys. Rev. Lett.* 92 (2004) 246401: Erratum, *ibid*, 95 (2005) 109902.
- [14] O. Gunnarsson and B. I. Lundqvist: *Phys. Rev. B* 13 (1976) 4274.
- [15] L. X. Benedict, N. G. Chopra, M. L. Cohen, A. Zettl, S. G. Louie, and V. H. Crespi: *Chem. Phys. Lett.* 286 (1998) 490.
- [16] Y. Baskin and L. Mayer: *Phys. Rev.* 100, (1955) 544.
- [17] H. Rydberg, M. Dion, N. Jacobson, E. Schroder, P. Hyldgaard, S. I. Simak, D. C. Langreth, and B. I. Lundqvist: *Phys. Rev. Lett.* 91 (2003) 126402.
- [18] T. Thonhauser, Valentino R. Cooper, Shen Li, Aaron Puzder, Per Hyldgaard, and David C. Langreth: *Phys. Rev. B* 76 (2007) 125112.