文部科学省次世代ＩＴ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.11.00

チュートリアル・ガイド

■ お問い合わせ先

(公開/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

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 URL http://www.ciss.iis.u-tokyo.ac.jp

■ 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 フリーソフトウエア使用許諾条件

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使用できる行為には、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

図 目 次

1 はじめに

本プログラムの名称は PHASE です。

PHASE は，物質中の電子の状態を，密度汎関数理論に基づいて計算します。本チュートリアルでは，第 3 節以降に設けた例題をもとに，ユーザーの皆様に簡単な計算を体験していただき，PHASE の使用法を学んでいただきます。詳細な使い方は，別冊のユーザーマニュアルでご説明いたします。第 3 節に進む前に，PHASE がどの様な機能・特徴を持つのかという点についてご説明いたします。また，第 2 節では，PHASE のインストールの仕方をご説明します。

1.1 PHASE で何が計算できるか？

PHASE を使うと，物質の持つ以下のような性質が，理論的に求められます。

1. 全エネルギー
2. 電荷密度の分布
3. 電子の状態密度
4. バンド構造
5. 安定な原子構造
6. 振動解析
7. STM 像シミュレーション

1.2 PHASE の特徴

1. Local Density Approximation (LDA) や Generalized Gradient Approximation (GGA) を使って，密度汎関数法に基づいた電子状態の計算が可能です。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おり，非常に信頼性の高い計算法であると言えます。
2. 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します。使用する擬ポテンシャルとしては，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あります。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，大規模計算が可能になります。
3. 波動関数の最適化には，残差最小化 (RMM) 法，Steepest Descent (SD) 法，改良型 SD 法，一次元探索法などが選択できます。また，電荷密度の混合には，単純混合法，Broyden 法などが使用できます。
4. 幾何学的構造の最適化法としては，Quenched MD 法と GDIIS 法, CG 法が使えます。
5. 入力ファイルがタグ形式で構成されています。これにより，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なっています。
6. バンド構造や状態密度 (DOS) を描画する Perl スクリプトが付属しています。「革新的シミュレーションソフトウエアの研究開発」プロジェクトで開発された Biostation viewer を用いれば，結晶構造や電荷密度の三次元表示が可能です。
7. PHASE は MPI を使用して並列化されています。

このように， PHASE を使うと，豊富に用意された最適化方法を組み合わせることにより，安定で効率の良い，原子構造や電子状態の最適化が行なえます。さらに，付属の描画ソフトを用いることにより，計算結果の解析も容易に行なえます。

2 インストール・ガイド

2.1 パッケージの展開

最初に，PHASE のプログラムを展開します。
Linux や FreeBSD などの，GNU のフリーソフトがインストールされたシステムをお使いの場合には，

```
% tar zxvf phase_v1100.tar.gz
```

それ以外の UNIX マシンの場合には，

```
% gunzip phase_v1100.tar.gz
% tar -xvf phase_v1100.tar
```

として，phase_v1100.tar.gz を展開して下さい。ここで，

```
%
```

と書いたのは，シェルプロンプトのことです。
これらの解凍の作業で，

1. doc (含：phase_install.pdf, phase_tutorial.pdf, phase_samples.pdf)
2. bin
3. src_phase
4. src_stm
5. tools
6. samples

という 6 個のディレクトリーが作られたはずです。

2.2 PHASE のインストール

PHASE のインストールは、インストーラ intall.sh を用いて行います。

```
% cd $HOME/phase_v1100
% ./install.sh
```

詳しい説明はインストールマニュアル phase_install.pdf に書かれていますので、ご参照ください。

2.3 テスト計算

ここでは，念のためにテスト計算を行なってみます。
PHASE のサンプル・ファイルの中で，計算に必要な原子数が最小である，バルクのシリコンについて計算をしてみます。
第 2.1 節で phase_v1100.tar.gz を展開したディレクトリーで，

```
% cd phase/samples/testrun
```

として，testrun という名前のディレクトリーに移って下さい。このディレクトリーで，

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

とタイプしてリターンキーを押すと，テスト計算が始まります。マシンにもよりますが，数分程度でこの計算は終るはずです。何も異常が表示されないで計算が終了したら，インストールが無事に済んだ証拠です。テスト計算の結果，output000 というファイルが作られているはずですので，

```
% grep TOTAL output000
```

として得られた結果を，以下と比較してみて下さい。

```
TOTAL ENERGY FOR     1 -TH ITER=     -7.858496652840   edel =  -0.785850D+01
TOTAL ENERGY FOR     2 -TH ITER=     -7.877642072374   edel =  -0.191454D-01
TOTAL ENERGY FOR     3 -TH ITER=     -7.884173165960   edel =  -0.653109D-02
TOTAL ENERGY FOR     4 -TH ITER=     -7.894869062162   edel =  -0.106959D-01
TOTAL ENERGY FOR     5 -TH ITER=     -7.895854406233   edel =  -0.985344D-03
TOTAL ENERGY FOR     6 -TH ITER=     -7.896896464510   edel =  -0.104206D-02
TOTAL ENERGY FOR     7 -TH ITER=     -7.896965336639   edel =  -0.688721D-04
TOTAL ENERGY FOR     8 -TH ITER=     -7.896987810477   edel =  -0.224738D-04
TOTAL ENERGY FOR     9 -TH ITER=     -7.896999000751   edel =  -0.111903D-04
TOTAL ENERGY FOR    10 -TH ITER=     -7.897009945109   edel =  -0.109444D-04
TOTAL ENERGY FOR    11 -TH ITER=     -7.897012361067   edel =  -0.241596D-05
TOTAL ENERGY FOR    12 -TH ITER=     -7.897014072114   edel =  -0.171105D-05
TOTAL ENERGY FOR    13 -TH ITER=     -7.897014580162   edel =  -0.508047D-06
TOTAL ENERGY FOR    14 -TH ITER=     -7.897014905148   edel =  -0.324986D-06
TOTAL ENERGY FOR    15 -TH ITER=     -7.897015023768   edel =  -0.118620D-06
TOTAL ENERGY FOR    16 -TH ITER=     -7.897015098853   edel =  -0.750857D-07
TOTAL ENERGY FOR    17 -TH ITER=     -7.897015127853   edel =  -0.289999D-07
TOTAL ENERGY FOR    18 -TH ITER=     -7.897015145070   edel =  -0.172165D-07
TOTAL ENERGY FOR    19 -TH ITER=     -7.897015151163   edel =  -0.609355D-08
TOTAL ENERGY FOR    20 -TH ITER=     -7.897015154701   edel =  -0.353806D-08
TOTAL ENERGY FOR    21 -TH ITER=     -7.897015155674   edel =  -0.972411D-09
TOTAL ENERGY FOR    22 -TH ITER=     -7.897015156285   edel =  -0.611410D-09
TOTAL ENERGY FOR    23 -TH ITER=     -7.897015156328   edel =  -0.425100D-10
TOTAL ENERGY FOR    24 -TH ITER=     -7.897015156328   edel =  -0.509814D-12
TOTAL ENERGY FOR    25 -TH ITER=     -7.897015156328   edel =  -0.113687D-12
```

この比較の結果に，小数点以下 10 桁程度までの範囲で矛盾が見つからなければ，テスト計算は無事終了です。

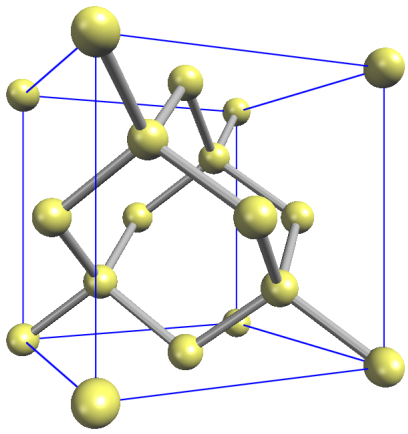

図 1: シリコン原子が構成するダイヤモンド構造

3 例題 1: シリコン結晶 (Si_8)

では，早速例題を始めることにしましょう。第 2.1 節で行なった phase_v1100.tar.gz の展開で作られた phase_v1100 というディレクトリーの下の，samples というディレクトリーに移ると，その中に幾つか例題用のディレクトリーが見つかるはずです。ここでは，シリコン原子 8 個からなる，Si_8 という系についての計算をおこなうことにします。図 1 に示したのは，シリコン原子からなるダイヤモンド構造です。

Si8 という名前のディレクトリーには，以下のファイルが格納されています。

```
file_names.data
input_scf_Si8.data
```

これらは入力に関するものです。

3.1 入力ファイルの説明

phase の計算に使う入力ファイル名は，file_names.data という名前のファイルの中で指定されています。このファイルの中を見ると，(!で始まるコメント文を除いて)

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

と書かれているはずです。Si_ldapw91_nc_01.pp はシリコンの擬ポテンシャル・データです。これらのファイルが PHASE を使った計算の過程でどの様に使われるかを，図 2 に示しました。

PHASE を走らせるためには，最低限，擬ポテンシャルデータ F_POT(1) と，標準入力ファイル F_INP が指定されている必要があります。前者は，CIAO を使用して作成することが可能ですが，本節の例題で行なう計算に必要な，シリコン原子用の擬ポテンシャル・データを格納したファイル Si_ldapw91_nc_01.pp が，このディレクトリー内にすでに用意されています。後者は，ユーザが，どの様な作業を PHASE に行なわせるかを指定するために必要な入力ファイルで，その中で使用されている各変数については，本例題に沿って説明します。

まず最初に， F_INP で指定されている入力ファイルである，input_scf_Si8.data について説明します。このファイルはタグ形式で記述されています。ここでは，計算上，特に重要な部分について説明します。このファイルの冒頭に指定されている第 1 のタグ

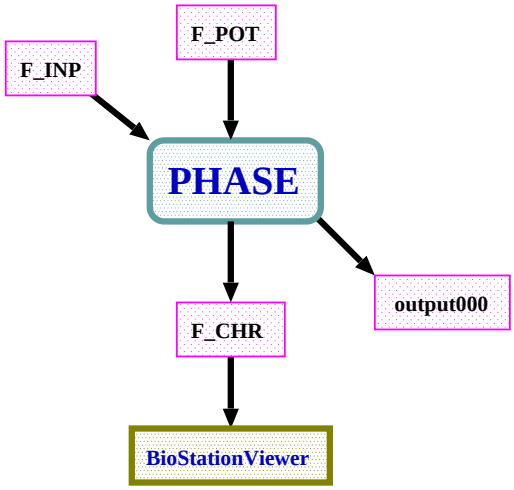

図 2: 例題 1： Si_8 の処理過程のフローチャート．実行ファイル PHASE に向かう矢印は入力を，逆は出力を意味する．また，青色と赤色はそれぞれ，実行ファイルと入出力ファイルを指している．F_INP (input_scf_Si8.data)：入力ファイル, F_POT (Si_ldapw91_nc_01.pp)：擬ポテンシャル, F_CHR (nfchr.cube)：電荷密度分布の出力ファイル, output000：計算結果の標準出力データ (ログファイル), Biostation viewer：「革新的シミュレーションソフトウエアの研究開発」プロジェクトで開発された描画用ソフト．

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

は，計算時間の最大値を指定しています。

その次にある第 2 のタグは，計算精度を指定するもので，それは以下のように記述されています。

```
accuracy{
        cutoff_wf =   9.00  rydberg
        cutoff_cd =  36.00  rydberg
        num_bands = 20
        ksampling{
                method = mesh ! {mesh|file|directin|gamma}
                mesh{  nx = 4, ny =  4, nz =  4  }
        }
        ...
        xctype = ldapw91
        scf_convergence{
             delta_total_energy = 1.e-12 hartree
             succession = 3      !default value = 3
        }
        ...
}
```

最初の 2 つのパラメーター cutoff_wf と cutoff_cd は，波動関数と電荷密度分布のカットオフ・エネルギーが，それぞれ 9.0Ry と 36.0Ry という値であることを表しています。

num_bands はエネルギー準位数を表します。この計算では，Si 原子 8 個を扱いますが，各原子は 4 個の価電子をもつため，占有される準位数は，スピンの縮退度を考慮すると 8×4/2=16 となります。このため num_bands は，16 以上に設定しておく必要があります。また，ksampling というタグは，$\boldsymbol{k}$ 点のサンプリングの方法を指定するのに使われます。この例では，$4 \times 4 \times 4$ のメッシュ点が標本点となります。

xctype = ldapw91 では，LDA 型交換相関エネルギーを指定しています。

その次のタグにある 2 つの変数は，計算の収束条件を表しています。この例の場合，全エネルギーの計算誤差が 10^{-12} Hartree 未満に収まるという結果が連続して 3 回続いたら，計算を終了させるように指定されています。

第 3 のタグである structure では，結晶構造が指定されます。単位はデフォルトが自然単位となっています（長さの単位は Bohr)。

```
structure{
        unit_cell_type = primitive
        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
        }
}

 #default weight = 1, element = Si, mobile = 1
```

が全ての原子に適用されています。

さらに，atom_list では，計算に使う原子種を指定し，その単位胞内での内部座標位置と，それぞれの原子の位置が固定されているか否かを指定しています。element_list では，元素名とその原子番号が指定されます。これに対応して，擬ポテンシャル・データが格納されたファイル Si_ldapw91_nc_01.gncpp2 の 1 行目には，

```
   14   4   3   0  : NATOMN, IVAL, ILOC, ITPCC
```

として，シリコンの原子番号 14，価電子数 4 の数字が記述されています。

入力ファイルの最後には，計算終了後の後処理用のタグが記述されています。

```
postprocessing{
        ...
        charge{
                sw_charge_rspace    = ON
                filetype            = cube  !{cube|density_only}
                title  = "This is a title line for the bulk Si"
        }
}
```

上記指定により，電荷密度が file_names.data 内で，F_CHR という変数で指定されるファイルに出力されます。filetype = cube とする事により，cube 形式で出力されます。このとき，F_CHR で指定される file の名前は，*.cube の形式である必要があります。cube ファイルは「革新的シミュレーションソフトウエアの研究開発」プロジェクトで公開している描画ソフト Biostation viewer を使ってグラフィックス表示することが可能です。

3.2 実行ファイル phase を使った計算の実行

ここで，全エネルギーを求めてみます。お使いの計算機上に，MPI がインストールされていることを確認し，

```
% 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ーの数，エネルギー準位の分割計算の数，および，$\boldsymbol{k}$ 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 × C という関係が成り立っていなければなりません。

また，1 CPU の計算機を使う場合には，

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

として下さい。この後，リターン・キーを押してジョブを投入します。

計算の途中経過を確認するには，計算のログ・ファイルである output000 に書き込まれている全エネルギーの計算結果が，暫時減少しているかどうかを見るのが，最も簡便な方法でしょう。2 ノード以上で並列計算を行なった場合には，output000_001 や output000_002 なども出力されますが，これらは別のノードの出力であり，並列デバッグ出力を行わない限り、空のファイルです。

ここでは，

```
% grep TOTAL output000
```

として得られた結果を列記します。Si8 のサンプルを使って得られた output000 に対して，上記の grep をおこなうと，次のような結果が表示されます。

```
TOTAL ENERGY FOR     1 -TH ITER=    -30.851502112276   edel =  -0.308515D+02
TOTAL ENERGY FOR     2 -TH ITER=    -31.428857832957   edel =  -0.577356D+00
TOTAL ENERGY FOR     3 -TH ITER=    -31.547875271353   edel =  -0.119017D+00
TOTAL ENERGY FOR     4 -TH ITER=    -31.575313743308   edel =  -0.274385D-01
TOTAL ENERGY FOR     5 -TH ITER=    -31.582591031973   edel =  -0.727729D-02
TOTAL ENERGY FOR     6 -TH ITER=    -31.585296287695   edel =  -0.270526D-02
TOTAL ENERGY FOR     7 -TH ITER=    -31.586566551584   edel =  -0.127026D-02
TOTAL ENERGY FOR     8 -TH ITER=    -31.587203940144   edel =  -0.637389D-03
TOTAL ENERGY FOR     9 -TH ITER=    -31.587536187844   edel =  -0.332248D-03
TOTAL ENERGY FOR    10 -TH ITER=    -31.587714367315   edel =  -0.178179D-03
TOTAL ENERGY FOR    11 -TH ITER=    -31.587811775875   edel =  -0.974086D-04
TOTAL ENERGY FOR    12 -TH ITER=    -31.587865777306   edel =  -0.540014D-04
TOTAL ENERGY FOR    13 -TH ITER=    -31.587896135394   edel =  -0.303581D-04
TOTAL ENERGY FOR    14 -TH ITER=    -31.587913347827   edel =  -0.172124D-04
TOTAL ENERGY FOR    15 -TH ITER=    -31.587923218322   edel =  -0.987050D-05
TOTAL ENERGY FOR    16 -TH ITER=    -31.587928921902   edel =  -0.570358D-05
TOTAL ENERGY FOR    17 -TH ITER=    -31.587932250599   edel =  -0.332870D-05
TOTAL ENERGY FOR    18 -TH ITER=    -31.587934208228   edel =  -0.195763D-05
TOTAL ENERGY FOR    19 -TH ITER=    -31.587935369846   edel =  -0.116162D-05
TOTAL ENERGY FOR    20 -TH ITER=    -31.587936064369   edel =  -0.694523D-06
TOTAL ENERGY FOR    21 -TH ITER=    -31.587937128483   edel =  -0.106411D-05
TOTAL ENERGY FOR    22 -TH ITER=    -31.587937146269   edel =  -0.177857D-07
TOTAL ENERGY FOR    23 -TH ITER=    -31.587937147223   edel =  -0.953783D-09
TOTAL ENERGY FOR    24 -TH ITER=    -31.587937147361   edel =  -0.138854D-09
TOTAL ENERGY FOR    25 -TH ITER=    -31.587937147369   edel =  -0.733991D-11
TOTAL ENERGY FOR    26 -TH ITER=    -31.587937147369   edel =  -0.358824D-12
TOTAL ENERGY FOR    27 -TH ITER=    -31.587937147369   edel =  -0.117240D-12
```

計算を繰り返す毎に，全エネルギーの値が収束してゆく様子が分かります。この計算は，第 3.1 節の入力ファイルに書かれた，全エネルギーに対する収束条件を満たしたところで終了しています。お使いの計算機でも，同じ結果が得られるかどうかを確認して下さい。

計算が終ったら，ファイル，nfchr.cube，が作られているはずです。「革新的シミュレーションソフトウエアの研究開発」プロジェクトで公開している描画ソフト Biostation viewer を用いて，原子構造図や，電荷密度分布の 3 次元表示を行うことができます。

ここでは，そのうちの電荷密度分布を図 3 に示します。

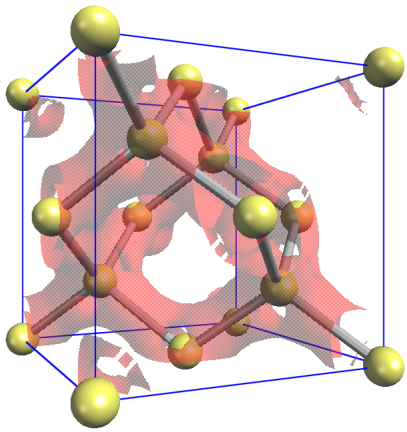

図 3: シリコン結晶の電荷密度分布 (原子数を増やすなど，cube file に修正を加えています)

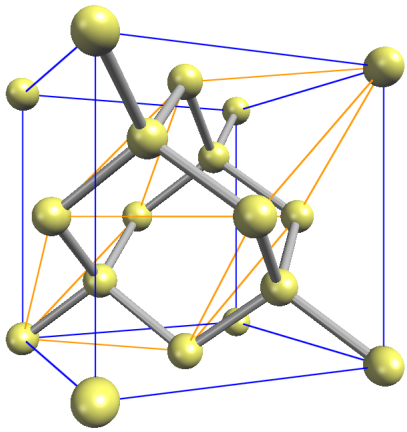

図 4: Si_2 の原子構造図。黄線は原子 2 個を含む基本格子を表す

4 例題 2 : バンド計算 – シリコン結晶 (Si_2)

今度は，バンド計算を行なってみましょう。シリコン原子が構成するダイヤモンド構造の基本格子は，原子 2 個を含むので，ここでは，シリコン原子 2 個からなる Si_2 という系を扱います。原子構造は図 4 に，また，本節で行なう作業の手順は図 5 に示しました。

4.1 SCF 計算

その後，例題が納められているディレクトリーである samples の中の Si2 というディレクトリーに入ります。そこで，scf というディレクトリーに入り，その中で，第 3 節と同様の手順に従って，phase を使った計算を行ない，全エネルギーの計算結果を収束させます。

ファイル file_names.data の中で，

```
F_INP    = './input_scf_Si.data'
F_POT(1) = '../../pp/Si_ldapw91_nc_01.pp'
  ...
```

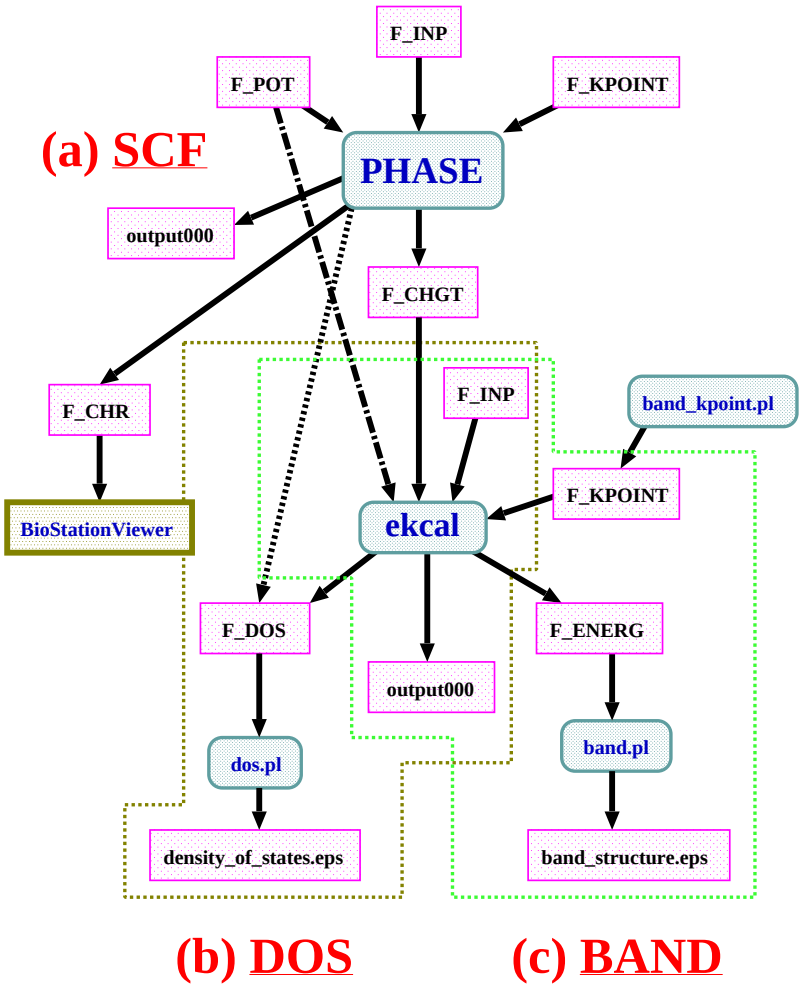

図 5: 例題 2 : Si_2 を使ったバンド計算の処理過程のフローチャート. (a) SCF 計算 (第 4.1 節), (b) 状態密度 (DOS) の計算（第 4.2 節）, (c) バンド構造図 (第 4.3 節). 実行ファイル PHASE に向かう矢印は入力を，逆は出力を指す. また，青色と赤色はそれぞれ，実行ファイルと入出力ファイルを指している. F_INP (input_scf_Si.data 等) : 入力ファイル, F_POT (Si_ldapw91_nc_01.pp) : 擬ポテンシャル, PHASE : 本プログラムの実行ファイル本体で，電荷密度分布の計算結果を F_CHGT (nfchgt.data) に出力. ekcal : F_CHGT を入力ファイルとして，第 4.2 節では状態密度の計算結果を F_DOS (dos.data) に出力。第 4.3 節では，同じ入力ファイルを使ってバンド計算を行ない，その結果を F_ENERG (nfenergy.data) に出力. band_kpoint.pl : $\boldsymbol{k}$ 点の入力データである F_KPOINT (kpoint.data) を生成する Perl スクリプト. dos.pl : F_DOS (dos.data) 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 $\boldsymbol{k}$ 点のエネルギーデータの出力ファイル F_ENERG (nfenergy.data) 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).

として指定されている入力ファイル input_scf_Si.data の，第 3 節で用いた入力ファイルとの違いは，取り扱う原子の数の違いによるものです。

```
accuracy{
        cutoff_wf =   9.00  rydberg
        cutoff_cd =  36.00  rydberg
        num_bands = 8
}

structure{
        unit_cell_type = Bravais
        unit_cell{
             a = 10.26, b = 10.26, c = 10.26
             alpha = 90, beta = 90, gamma = 90
        }
        symmetry{
             crystal_structure = diamond
        }

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

エネルギー準位数を表す num_bands の値は，2 個という原子数に応じて，前回よりも小さく取ってあります。格子ベクトルの大きさが前回の半分の値になっているのは，座標の取り方の違いによるものです。

この入力ファイルを使って，

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

として計算するわけですが，この計算が終わると，file_names.data というファイルの中で，変数 F_CHGT の入力値として指定されている出力ファイル nfchgt.data に，計算によって得られた電荷の情報が出力されます。

4.2 状態密度 (DOS) の計算

次に，

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

として，dos という名前のディレクトリーに移ります。こうして別ディレクトリーを利用するのは，波動関数の計算結果の出力ファイルである zaj.data が上書きされるのを避けるためです。

このディレクトリーにある制御用ファイル file_names.data では，入出力ファイル名が以下のように指定されています。

```
F_INP    = './input_dos_Si.data'
F_POT(1) = '../../pp/Si_ldapw91_nc_01.pp'
  ...          ...

F_CHGT   = '../scf/nfchgt.data'
  ...          ...

F_ENERG  = './nfenergy.data'
  ...          ...
```

入力ファイルは input_dos_Si.data と nfchgt.data の 2 つです。
入力ファイル input_dos_Si.data について，前節の計算で使った入力ファイル input_scf_Si.data と異なる部分を以下に示します。

```
Control{
       condition = fixed_charge
}

accuracy{
        cutoff_wf =   9.00  rydberg
        cutoff_cd =  36.00  rydberg
        num_bands = 8
        ksampling{
                method = mesh
                mesh{  nx = 4, ny =  4, nz =  4   }
        }
        smearing{
                method = tetrahedral
        }
        xctype = ldapw91
        initial_wavefunctions = matrix_diagon
            matrix_diagon{
               cutoff_wf =  9.00  rydberg
            }
        ek_convergence{
                num_max_iteration = 200
                sw_eval_eig_diff = on
                delta_eigenvalue = 1.e-8 hartree
                succession   = 2
       }
}

postprocessing{
       dos{
          sw_dos      = ON
          method      = tetrahedral   !{ tetrahedral | Gaussian }
          deltaE_dos  = 1.e-3 eV
          nwd_window_width = 10
       }
}
```

最初のタグである Control の部分で，phase による計算で得られた電荷の分布を固定することが指定されています。第 2 のタグでは，$\boldsymbol{k}$ 点の標本数を $4\times4\times4$ にして四面体法を用いることと，バンド計算に必要な ek_convergence の収束条件とを指定しています。最後のタグでは，計算終了後の後処理の過程で，四面体法による状態密度の計算結果の書き出しと，その際のエネルギーの精度が指定されています。
2 つめの入力ファイルである電荷密度のデータは，図 5 からも分かるように，第 4.1 節の計算で得られた出力ファイル nfchgt.data を使います。これは，入出力ファイルの制御用ファイル file_names.data の中で，

```
F_CHGT   = '../scf/nfchgt.data'
```

として指定されています。擬ポテンシャルも

```
 F_POT(1) = '../../pp/Si_ldapw91_nc_01.pp'
```

として，第 4.1 節と共通のものを使います。
これらの入力ファイルを使って状態密度の計算を行なうわけですが，ekcal は 1 プロセッサーのみに対応しているので，その計算は，

```
 % mpirun  ../../../bin/ekcal
```

として行ないます。
これでバンド計算が実行され，nfenergy.data という出力ファイルが生成されます。これは，各 $\boldsymbol{k}$ 点ごとのエネルギー値を，エネルギーの低い方から順に書き出したもので，その最初の部分は以下のようになっています。

```
 num_kpoints =    141
 num_bands   =      8
 nspin       =      1
 Valence band max   =   0.233846

=== energy_eigen_values ===
 ik =    1 (  0.000000  0.500000  0.500000 )
     -0.0484324491     -0.0484324491      0.1258095002      0.1258095002
      0.2619554320      0.2619554320      0.6015285289      0.6015285289
=== energy_eigen_values ===
 ik =    2 (  0.000000  0.490000  0.490000 )
     -0.0540717117     -0.0427149546      0.1258687813      0.1258687813
      0.2607026827      0.2633829946      0.6006244013      0.6006244013
=== energy_eigen_values ===
 ik =    3 (  0.000000  0.480000  0.480000 )
     -0.0596299923     -0.0369220783      0.1260465996      0.1260465996
      0.2596226501      0.2649874134      0.5980547648      0.5980547648
=== energy_eigen_values ===
 ik =    4 (  0.000000  0.470000  0.470000 )
     -0.0651046420     -0.0310567694      0.1263428799      0.1263428799
      0.2587131916      0.2667706685      0.5941566835      0.5941566835
=== energy_eigen_values ===
 ik =    5 (  0.000000  0.460000  0.460000 )
     -0.0704931128     -0.0251220735      0.1267574962      0.1267574962
      0.2579721226      0.2687346642      0.5892968047      0.5892968047
```

最初の 2 行は，それぞれ，$\boldsymbol{k}$ 点とバンドの数を表します。3 行目は，この計算でスピン分極は考慮されていないことを，また，4 行目は価電子帯上端におけるエネルギーの値を指しています。
次に，ekcal による計算結果の解析用 Perl スクリプトである dos.pl を使って，電子状態密度の図を描くことにします。描画するエネルギー範囲の最小値 E1 と最大値 E2 を決めて，

```
 % dos.pl dos.data -erange=E1,E2
```

とすると，Postscript 形式の状態密度図 density_of_states.eps が得られます。また，-with_fermi というオプションをつけて，この処理を実行すると，生成される状態密度図にフェルミ・レベルが点線で描かれます。ただし，ギャップのある系では，価電子帯のエネルギー最大値のところに点線が引かれます。
ここでは，

```
 % dos.pl dos.data -erange=-13,5 -with_fermi
```

として得られた状態密度を，図 6 に示します。
dos.pl の動作環境や詳細な使用法については，tools のマニュアルをご参照下さい。

4.3 バンド構造図

今度は band という名前のディレクトリーに移ります。

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

ここに格納されているファイル file_names.data では，以下のように，

```
 F_INP    = './input_band_Si.data'
 F_POT(1) = '../../pp/Si_ldapw91_nc_01.pp'
 F_KPOINT = '../tools/kpoint.data'
```

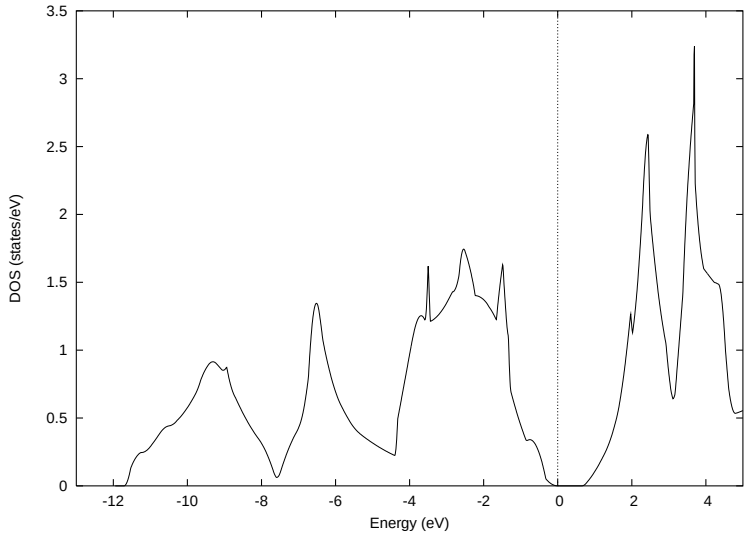

図 6: ekcal による計算で得られたシリコンのダイヤモンド構造の状態密度

```
F_CHGT   = '../scf/nfchgt.data'
  ...          ...
```

入力ファイルとして input_band_Si.data を使用することと，$\boldsymbol{k}$ 点のデータを kpoint.data から入力することが指定されています。

入力ファイル kpoint.data は，Perl スクリプト band_kpoint.pl と bandkpt_fcc_xglux.in という補助ファイルを使って，

```
% band_kpoint.pl bandkpt_fcc_xglux.in
```

として生成することができます。

また，Si_ldapw91_nc_01.pp と nfchgt.data は第 4.2 節同様，それぞれ，pp と scf というディレクトリーにあるものを使います。

これらの入力ファイルを使って，再び ekcal の計算を行ないます。

```
% mpirun  ../../../bin/ekcal
```

こうして得られた出力ファイル nfenergy.data を基に，Perl スクリプト band.pl を用いて，

```
% band.pl nfenergy.data bandkpt_fcc_xglux.in -erange=E1,E2 -with_fermi
```

としてデータ処理することにより，Postscript 形式のバンド構造図である band_structure.eps を出力します。ここでは，描画するエネルギー範囲の最小値 E1 と最大値 E2 を，以前同様 E1 = -13 と E2 = 5 として得られた結果を，図 7 に示します。

5 例題 3：原子構造の最適化

これまでの計算では，安定な原子構造を最初から指定して，それに応じた電子状態の計算を行なって来ました。本節では，わざと安定な原子配置から原子位置をずらして，そこからの緩和過程を計算してみます。再び Si2 のディレクトリーに入り，さらにその下にある relax という名のディレクトリーに移ります。

```
% cd Si2/relax
```

あるいは，前節のバンド構造の計算用のディレクトリーからだと，

```
% cd ../relax
```

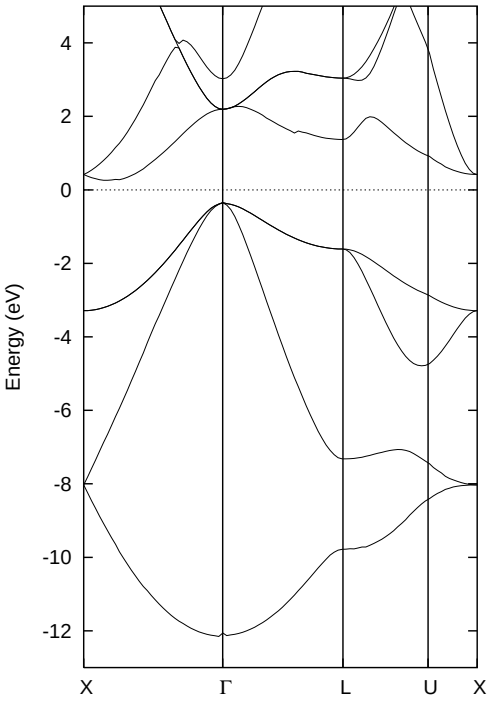

図 7: ekcal による計算で得られた Si_2 のバンド構造

となります。
さて，ここにある制御用ファイル file_names.data の中では，

```
F_INP    = './input_relax_Si.data'
  ...

F_DYNM   = './nfdynm.data'
  ...
```

として，入力ファイル input_relax_Si.data と，原子の位置座標と各原子に働く力の計算結果の出力ファイル nfdynm.data とが指定されています。この入力ファイルの，第 4 節で用いた scf ディレクトリーにある input_scf_Si.data という入力ファイルとの主な違いは，

```
structure{
         ...

        atom_list{
             atoms{
                #tag     rx         ry         rz     element mobile
                       0.130      0.130      0.130      Si    yes
                      -0.130     -0.130     -0.130      Si    yes
             }
       }
}
```

として，格子間隔を 0.125 から 0.130 に変えていることです。また，mobile 変数の値を yes にして，原子位置を可変にしています。
原子に働く力の収束条件は，同じ入力ファイルの中の計算精度のタグの部分で，

```
accuracy{
        force_convergence{
```

```
                max_force = 1.0e-3
        }
}
```

として与えられています。

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

として計算して得られる出力ファイル nfdynm.data は以下の通りです。

```
#
#   a_vector =         0.0000000000        5.1300000000        5.1300000000
#   b_vector =         5.1300000000        0.0000000000        5.1300000000
#   c_vector =         5.1300000000        5.1300000000        0.0000000000
#   ntyp =        1 natm =        2
# (natm->type)     1    1
# (speciesname)     1 :   Si
#
 cps and forc at (iter_ion, iter_total =     1      34 )
    1    1.333800000    1.333800000    1.333800000   -0.010794   -0.010794   -0.010794
    2   -1.333800000   -1.333800000   -1.333800000    0.010794    0.010794    0.010794
 cps and forc at (iter_ion, iter_total =     2      53 )
    1    1.331707297    1.331707297    1.331707297   -0.010402   -0.010402   -0.010402
    2   -1.331707297   -1.331707297   -1.331707297    0.010402    0.010402    0.010402
 cps and forc at (iter_ion, iter_total =     3      75 )
    1    1.327597870    1.327597870    1.327597870   -0.009614   -0.009614   -0.009614
    2   -1.327597870   -1.327597870   -1.327597870    0.009614    0.009614    0.009614
 cps and forc at (iter_ion, iter_total =     4     100 )
    1    1.321624355    1.321624355    1.321624355   -0.008433   -0.008433   -0.008433
    2   -1.321624355   -1.321624355   -1.321624355    0.008433    0.008433    0.008433
 cps and forc at (iter_ion, iter_total =     5     127 )
    1    1.314015753    1.314015753    1.314015753   -0.006865   -0.006865   -0.006865
    2   -1.314015753   -1.314015753   -1.314015753    0.006865    0.006865    0.006865
 cps and forc at (iter_ion, iter_total =     6     155 )
    1    1.305076108    1.305076108    1.305076108   -0.004930   -0.004930   -0.004930
    2   -1.305076108   -1.305076108   -1.305076108    0.004930    0.004930    0.004930
 cps and forc at (iter_ion, iter_total =     7     184 )
    1    1.295180554    1.295180554    1.295180554   -0.002671   -0.002671   -0.002671
    2   -1.295180554   -1.295180554   -1.295180554    0.002671    0.002671    0.002671
 cps and forc at (iter_ion, iter_total =     8     213 )
    1    1.284767108    1.284767108    1.284767108   -0.000159   -0.000159   -0.000159
    2   -1.284767108   -1.284767108   -1.284767108    0.000159    0.000159    0.000159
```

このうち，# 記号で始まる部分は入力データの一部を表していますが，その次の行は，イオンすなわちコア原子の位置座標を一回更新する間に，全更新回数が 34 回であったこと，すなわち，この間に波動関数が 33 回更新されたことを示しています。波動関数の更新に対する収束条件は，第 3 節の例題と同様に，全エネルギーに対して課されています。

また，その次の 2 行は，原子の番号，原子位置 (x,y,z, bohr 単位)，および力の成分 (x,y,z, hartree/bohr 単位) の計算結果を表しています。これにより，結果を下まで辿ってゆくと，計算が進むにつれて，原子に働く力が急激に減少してゆくことが分かります。最後の更新で，力の各成分の計算結果が，最初に指定された収束条件以下になったために，緩和過程の計算が終了しています。

6 例題 4 : 振動解析 - シリコン結晶 (Si_2)

6.1 入力ファイルの説明

この節では振動解析の仕方を説明しますが，例題 2 の SCF 計算の節の内容を理解してることを前提とします。
まず，シリコン結晶の振動解析を行うために，サンプルディレクトリの phonon/Si2 というディレクトリーに移ります。

```
cd phase/samples/phonon/Si2
```

ここに格納されている nfinput.data では element_list

```
element_list{     #units atomic_mass
                 #tag element  atomicnumber mass
                             Si       14     28.0855
        }
```

にシリコン原子の質量 28.0855 amu が指定されています。質量の単位を原子質量単位とすることを#units の後に atomic_mass を記述することで指定しています。
この入力には振動解析を制御する Phonon ブロックがあります。

```
Phonon{
   sw_phonon = on
   sw_calc_force = on
   displacement = 0.1
   sw_vibrational_modes = on
}
```

sw_calc_force と sw_vibrational_modes がともに ON になっているので，振動解析のための力計算を行い，それがすべて完了したあとに，振動解析が行われます。
SCF 計算行ったときと同じ様にして，PHASE を実行してください。

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

この計算が終わると，出力ファイル mode.data に振動解析の結果が出力されます。mode.data の最初の部分は以下のようになっています。

```
 --- primitive lattice vectors ---
   0.0000000000   5.0875600000   5.0875600000
   5.0875600000   0.0000000000   5.0875600000
   5.0875600000   5.0875600000   0.0000000000
 --- Equilibrium position and mass of each atom---
 Natom=    2
     1   1.2718900000   1.2718900000   1.2718900000    51196.42133 Si
     2  -1.2718900000  -1.2718900000  -1.2718900000    51196.42133 Si
 --- Vibrational modes ---
 Nmode=    6 Natom=    2
 n=     1 T1u
     hbarW=  0.00000000E+00 Ha =  0.00000000E+00 eV; nu=  0.00000000E+00 cm^-1
     1   0.0000000000  0.0000000000  0.7071067812
     2   0.0000000000  0.0000000000  0.7071067812
```

最初の二行目から三行目は基本並進ベクトルをあらわしています。六行目は原子数を表しています。その次の行からは，原子の番号，デカルト座標，質量，ラベルが一行にあらわされています。Vibrational modes というタイトル行の次の行にはモード数と原子数があらわされています。これ以降には各振動モードの既約表現を先頭行として，次行に振動数があらわされ，その次の行から固有ベクトルがあらわされています。固有ベクトルは原子の番号の後にその原子に帰属するベクトルの 3 成分があらわされています。

6.2 振動数レベル図

振動解析の出力ファイル mode.data から振動数の情報を取り出し，図にして見ましょう。

```
% freq.pl mode.data
```

とすると，Postscript 形式の振動数レベル図 freq.eps が得られます。これを図 8 に示します。この図から振動数が 517 cm^{-1} であるモードがあることが分かります。このモードの既約表現は T_{2g} であるので，同じ振動数のモードが三重に縮重しています。T_{2g} モードはラマン活性であり，図には既約表現の記号の右側に記号 R でそのことが示されています。赤外活性である場合には IR と示されます。

6.3 振動モードの可視化

BioStationViewer を使用すると，固有ベクトルを矢印表示したり，原子が振動するアニメーションとして振動モードを可視化したりできます。それを行うには，まず振動解析の出力ファイル mode.data から振動数の情報を取り出し，拡張 Trajectory 形式のファイル (拡張子:tr2) を作成するスクリプト animate.pl を使用します。

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

とすると，拡張 Trajectory 形式のファイルがモードの数だけ出力されます。control.inp には以下のように切り出すセルの原点とベクトルが指定されています。

```
origin  1.27189 1.27189 1.27189
vector1 10.17512 0 0
vector2 0 10.17512 0
vector3 0 0 10.17512
```

この入力では切り出すセルをブラベー格子の単位胞にとり，セルの原点にシリコン原子がくるように設定しています。たとえば，出力された拡張 Trajectory 形式のファイル mode_6.tr2 を BioStationViewer で表示させると，図 9 のように固有ベクトルが矢印で示されます。また，BioStationViewer の Trajectory 機能により，原子が振動する様子を見ることができます。図 9 に示されているセルは，出力された grid.mol2 ファイルを読み込ことで表示できます。

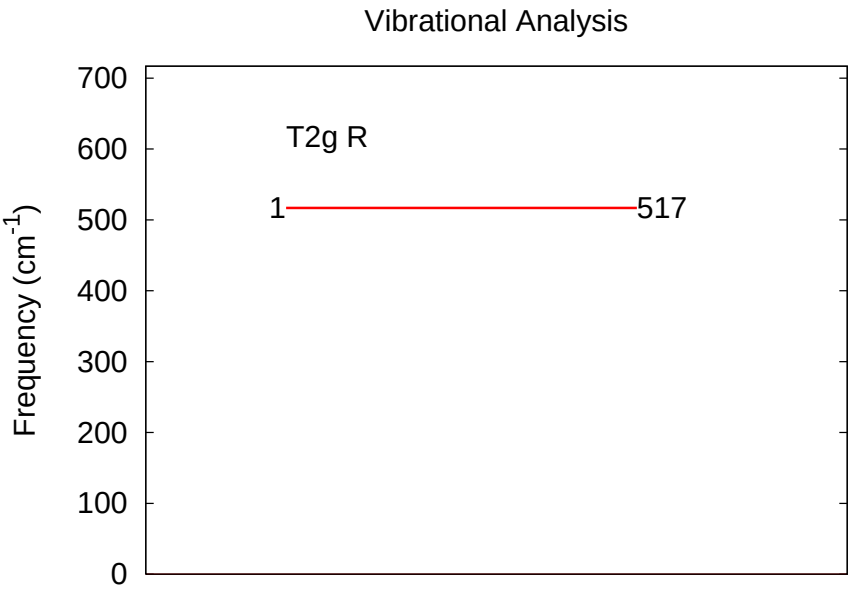

図 8: バルク Si の領域中心フォノンモードの振動数

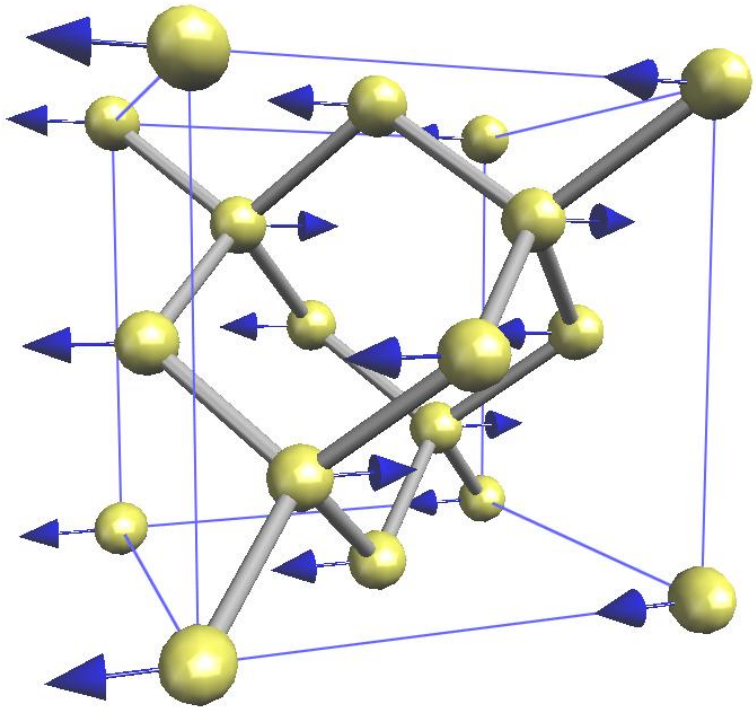

図 9: バルク Si の領域中心フォノンモードの固有ベクトル

7 例題 5: ストレステンソルと弾性定数 (Si)

この節ではストレステンソルと弾性定数を求めてみましょう。尚、問題を簡単にするため、本節では立方晶のみを扱います。より一般的な結晶構造をもつ系で計算した場合、本節で与えられる式と係数をそのまま用いることは出来ませんのでご注意ください。

7.1 入力ファイル

必要な入力ファイルは前節までと同様、file_names.data と file_names.data の中で指定してある擬ポテンシャルファイル、Si_scf.in です。ストレステンソルの計算に必要な変更は Si_scf.in だけです。
それでは、計算を実行するディレクトリへ移りましょう。

```
% cd PATH_TO_WORKING_DIRECTORY_FOR_STRESS
```

使用する Si_scf.in の内容は以下の通りです。

```
 1:Control{
 2:  cpumax = 24 hour
 3:}
 4:
 5:accuracy{
 6:  cutoff_wf =  20.25  rydberg
 7:  cutoff_cd =  81.00  rydberg
 8:  num_bands = 20
 9:  xctype = ggapbe
10:  ksampling{
11:    method = mesh
12:    mesh{ nx = 8, ny = 8, nz = 8 }
13:  }
14:  smearing{
15:    method = tetrahedral
16:  }
17:  scf_convergence{
18:    delta_total_energy = 1.0e-10  hartree
19:    succession = 3
20:  }
21:  force_convergence{
22:    delta_force = 1.0e-4
23:  }
24:  initial_wavefunctions = matrix_diagon
25:  matrix_diagon{
26:    cutoff_wf = 5.00  rydberg
27:  }
28:  initial_charge_density = Gauss
29:}
30:
31:structure{
32:  unit_cell_type = primitive
33:  unit_cell{
34:    #units angstrom  ! Unit of LENGTH changes to Angstrom.
35:    a_vector =   0.0000000000   2.7296850000   2.7296850000
36:    b_vector =   2.7296850000   0.0000000000   2.7296850000
37:    c_vector =   2.7296850000   2.7296850000   0.0000000000
38:  }
39:  symmetry{
40:    crystal_structure = diamond
41:  }
```

```
42:  atom_list{
43:    coordinate_system = internal
44:    atoms{
45:      #tag     rx       ry       rz    element  mobile  weight
46:           0.125   0.125   0.125         Si      yes       1
47:          -0.125  -0.125  -0.125         Si      yes       1
48:    }
49:  }
50:  element_list{ #tag   element   atomicnumber   dev
51:                           Si            14    1.2
52:  }
53:}
54:
55:structure_evolution{
56:  stress{
57:    sw_stress=1
58:  }
59:}
```

ここで、赤文字で示してある部分 (55 行目から 59 行目まで) がストレステンソルを計算する為の必須タグです。また、上記リスト左部分の数字 (n:) は本 tutorial で便宜上付けたものであり、実際のファイルに記述してはいけません。

これまでの SCF 計算と同様にして PHASE を実行します。

```
% mpirun PATH_TO_PHASE &
```

ここでは、比較的長い計算時間を要しますので、「&」を付けてバックグラウンドで実行しています。

計算が終了したら結果を確認しましょう。

```
% grep -A3 'STRESS TENSOR$' OUTPUT_FILENAME
  STRESS TENSOR
        0.0000003475         0.0000000000         0.0000000000
        0.0000000000         0.0000003475         0.0000000000
        0.0000000000         0.0000000000         0.0000003475
```

ここで得られたストレステンソルは

$$\begin{pmatrix} X_x & X_y & X_z \\ Y_x & Y_y & Y_z \\ Z_x & Z_y & Z_z \end{pmatrix} \tag{1}$$

の形式で出力されています。出力単位は [Hartree/Bohr^3] です。上の結果では入力データとして僅かに格子定数を小さく取ってあるため、正の X_x, Y_y, Z_z が出力されています。構造最適化などにより、原子間隔が釣り合いの位置にある時はストレステンソルの各成分は 0 になります。

また、釣り合いの位置からの格子変形 ($\equiv e$)、スティフネス定数 ($\equiv c$) を用いると次のようなフックの法則が成り立ちます。

$$\left.\begin{aligned} X_x &= c_{11}e_{xx} + c_{12}e_{yy} + c_{12}e_{zz} \\ Y_y &= c_{12}e_{xx} + c_{11}e_{yy} + c_{12}e_{zz} \\ Z_z &= c_{12}e_{xx} + c_{12}e_{yy} + c_{11}e_{zz} \\ X_y(=Y_x) &= c_{44}e_{xy} \\ Y_z(=Z_y) &= c_{44}e_{yz} \\ Z_x(=X_z) &= c_{44}e_{zx} \end{aligned}\right\} \tag{2}$$

7.2 弾性定数

さて、ストレステンソルの計算が出来ましたので、弾性定数を求めてみましょう。その為には、まず、ストレステンソルが 0 となるような格子定数を求めなければなりません[1]。すなわち、前節で示した入力ファイルの 35 行目から 37 行目を調節します。

[1]必要条件ではありませんが、正しくストレステンソルが計算されている事を確認する意味でも精密な格子定数を求められるようお勧めします。

ここでは、ストレステンソルが (ほぼ) 0 となるような格子定数を与えることにしますが、実際の計算では、PHASE 利用者自身によって探すことになります。

```
35:    a_vector =   0.0000000000   2.7297895000   2.7297895000
36:    b_vector =   2.7297895000   0.0000000000   2.7297895000
37:    c_vector =   2.7297895000   2.7297895000   0.0000000000
```

格子定数を上のように書き換え、前節と同様にしてストレステンソルを計算すると以下のような結果が得られるはずです。

```
% grep -A3 'STRESS TENSOR$' OUTPUT_FILENAME
  STRESS TENSOR
        0.0000000000         0.0000000000         0.0000000000
        0.0000000000         0.0000000000         0.0000000000
        0.0000000000         0.0000000000         0.0000000000
```

各成分が十分小さくなっていること (あるいは 0 になっていること) を確認して下さい。次に、x 軸方向だけを僅かに (ここでは 0.005 angstrom)[2]ずらします。

```
35:    a_vector =   0.0000000000   2.7297895000   2.7297895000
36:    b_vector =   2.7347895000   0.0000000000   2.7297895000
37:    c_vector =   2.7347895000   2.7297895000   0.0000000000
```

再度、格子定数を上のように書き換えます。また、この時には 39 行目から 41 行目までの `symmetry` タグは**必ず**消して (あるいはコメントアウトして) ください。編集を終えた上で計算した結果は以下のようになります。

```
% grep -A3 'STRESS TENSOR$' OUTPUT_FILENAME
  STRESS TENSOR
       -0.0000093954         0.0000000063         0.0000000016
        0.0000000063        -0.0000033142         0.0000000000
        0.0000000016         0.0000000000        -0.0000033163
```

ここで、本計算では回転や剪断ひずみを与えていませんので、非対角項は理論的に 0 となるべきところです。従って、上の結果で現れている非対角項は数値誤差によるものと考えられます。また、対称性により、Y_y 成分と Z_z 成分も一致すべきところですので、以下弾性定数の見積りに際しては Y_y, Z_z それぞれの平均値 (-0.00000331525) を用いることにします。

計算で用いた格子定数と釣り合いの位置からの変形 (ここでは x 軸方向へ 0.01 angstrom)、得られたストレステンソルを式 (2) へ代入すると、スティフネス定数 (c_{11}, c_{12}) がそれぞれ

$$\left.\begin{aligned} c_{11} &= 1.5091525 \\ c_{12} &= 0.5325178 \end{aligned}\right\} \tag{3}$$

のように求まります (単位は $[10^{12}\mathrm{dyn/cm^2}]$)。

一方、弾性定数 (ヤング率 ($\equiv Y$)・ポアソン比 ($\equiv P$)・体積弾性率 ($\equiv V$)) はスティフネス定数を用いて次のような式で書き表されます[3]。

$$\left.\begin{aligned} Y &= \frac{c_{11}^2 + c_{11}c_{12} - 2c_{12}^2}{c_{11} + c_{12}} \\ P &= \left|\frac{c_{12}}{c_{11} + c_{12}}\right| \\ V &= \frac{c_{11} + 2c_{12}}{3} \end{aligned}\right\} \tag{4}$$

これに式 (3) の結果を代入すれば Si の弾性定数が以下のように求まります。

$$\left.\begin{aligned} Y &\approx 1.231[10^{12}\mathrm{dyn/cm2}] = 123.1[\mathrm{GPa}] \\ P &\approx 0.261 \\ V &\approx 0.858[10^{12}\mathrm{dyn/cm2}] = 85.8[\mathrm{GPa}] \end{aligned}\right\} \tag{5}$$

以上でストレステンソルおよび弾性定数の説明は終りです。

より精度の高い弾性定数の計算を行ないたい場合、`cutoff_wf, cutoff_cd` を大きめにとり、電子状態を充分に収束させなければなりません。他の例題に比べ時間の掛かる計算になりますが、是非チャレンジしてみて下さい。

[2]unit cell としては x 軸方向に 0.01 angstrom 大きくなります。
[3]剛性率は $Y/(2+2P)$ と書けます。

8 おわりに

これで PHASE のチュートリアルは終了です。実際にご自分の研究に必要な計算を遂行する際には，PHASE, CIAO，PHASE TOOLS のユーザーズ・マニュアルもご参照下さい。