First-Principles Electronic-State Calculation Code by Pseudopotential Method Osaka2002_*nano*

VOL. I

– Basic Usage –

First Edition

Koun SHIRAI

ISIR, Osakak University

2 November 2004

History

Rev. 0.7 04 March 2003
Rev. 0.9 18 December 2003
Rev. 1.0 04 February 2004 Rev. 1.1 02 November 2004

Contents

Introduction

A package "Osaka2002_nano" (or shortly "Osaka2002" or Osaka2k) is a set of program codes which calculate electronic structures of materials by first-principles pseudopotential method. It covers a wide range of calculations from optimization of crystal structure to molecular dynamic simulations, in addition to standard self-consistent calculation and band calculations. Every components are the art-of-state calculations.

What we call nowadays first-principles calculations are also called in many ways, *e.g.*, *ab-initio* calculation, parameterless calculation, *etc.* The naming is cynical. [1] Which principle is the first one? The approach is not *a posteriori* (empirical?), but really heavy experiences are needed to acquaint the skill of calculations. Parameterless is by no means no parameters in calculations. Although no parameter fitted to experiment is assumed, there are indeed a couple of control parameters of calculations, which may yield different answers by the input value. The statement that only atomic numbers are required as input is merely a slogan of first-principles researchers. Actually, serious working experience is needed. This makes beginners to hesitate to work on.

This manual is written to alleviate the barriers and pains which beginners suffer. The intended level of this manual is such one that beginners can acquaint skills of professional calculations by self study alone. Special attention is paid that not specialists but experimentalists use this package in order to interpret their obtained data.

I should mention in advance that this English edition is not complete translation from the original Japanese text. Owing to the authors limited

[1] This is average Japanese impression to the name, first-principle*s*. I don't know how natural these naming sounds to Western people. Linguistically speaking, plural form is used only when special meaning is added. Furthermore, it is difficult to accept such an idea that there are many first things. Here is a reason why Japanese is easily disrupt to distinguish between countable and uncountable nouns.

ability of English, some parts are omitted, and the English text was not well polished out. I solicit the readers generosity in this regard.

History

As in many other programs, Osaka2k is also not created by one person, despite that only one person is indicated as the author in this manual. Here, by writing the names of contributors, I would like to express my acknowledgments to those pioneers.

A program `atom` which generates atomic pseudopotentials is created by Troullier and Martins [10], which itself has the long history. Osaka2k uses it as it were.

Development of the core program of Osaka2k is dated back to 1987 at Tohoku University. There, under the direction of Prof. Katayama-Yoshida, Dr. N. Orita (Now, AIST) as the primary writer and Dr. T. Sasaki (NIRIM), and T. Nishimatsu (Tohoku Univ.) developed a first-principles molecular dynamic simulation program, named `cpgs` at that time, which based on Car-Parrinello method. The primal use of `cpgs` was study of impurities in semiconductors at that time, and for this purpose, `cpgs` had been completed. Therefore, they are really the parents of this program. `cpgs` had been developed there until 1995.

However, the purpose of `cpgs` was limited, several deficients were found, such as only $\Gamma-$point sampling, no use of symmetry, etc. After Prof. Katayama-Yoshida moved to Osaka Univ. at 1995, the present author began to rewrite it in order to use of crystal symmetry fully, and multi k sampling. In this process, the author reconstructed the code in order to make use of `TSPACE` [24] created by Prof. A. Yanase. In addition, the core part of SCF calculation was replaced with the method of Teter-Payne-Allan [18], and functions of atomic optimization etc had been added.

At the same time, band, DOS, phonon calculations were developed by the author. Until the fall of 2000, all the components had been integrated all together. This was the original form of Osaka2000, which was opened to public at 2000.

After that, the whole codes were rewritten from beginning by new Fortran 90 under unified direction of programming design. This yielded a completely new version "Osaka2002_nano"

legal matters

As usual, we cannot owe any social responsibility for consequences of using this program package. The correctness of obtained result is after all users' responsibility, even though we paid our best efforts for the program codes to be correct.

When you publish papers, which use data obtained by "Osaka2k", please cite Osaka2k in text or references. "Osaka2k" is of course just a name of a program, so that readers cannot understand which method is used, if only this name is written. In this case, some of the original papers, which Osaka2k is based on, should be cited. To what extent should be cited is kind of tast, but when you do not have any idea about this, we recommend the following two papers.

As to used potentials, Troullier-Martins [10], As to calculational method, a review paper by Payne et. al. [19].

As to graphics of band diagram, there is no need to say anything when elemental drawing like 7.5 is used. But when using high-quality graphics like Fig. 7.6, you should refer program `ayband` created by A. Yanase.

source codes

The source codes can be obtained from the following site,

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

When you find program bug or errors in the manual, please send messages to

`koun@sanken.osaka-u.ac.jp`

Chapter 1

THEORY

In this section, the underlying principles of Osaka2k are described. Though it is more appropriate to study the basics of densiy functional theory (DFT) by standard textbooks, some conceptions and technicals are explained in some length here, because especially those topics written in paragraphs 1.7 and 1.5 are rarely treated in usual textbooks.

1.1 Density functional method

Theoretically, the properties of solid can be obtained by solving the eigenstates of total Hamiltonian of the system. In the atomic unit system, nonrelativistic Hamiltonian of the system is given by [1]

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

where $i-$ and j-summations take over all electron positions, $\mathbf{R}_n$ nth-atom position、$V_{\rm ion}(\mathbf{r}_i)$ atom potential at $\mathbf{r}_i$, and $E_{\rm ion}(\{\mathbf{R}_n\})$ the ion-ion direct interaction.

Most of the properties of the system being in interest such as the total energy of the ground state, atom force, electron density and electrostatic potential, etc. can be obtained by solving Schrödinger equation:

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

[1] Usually, e^2 is set to 2 in the atom unit system, however, there are Rydberg unit and Hartree unit even if it is named atomic units, to avoid this confusion we left e^2as it was here.

One of attempts of non-empirical method to obtain the properties of solid is to solve the equation of the many-electron Hamiltonian (1.1) directly. In practice, the equation (1.2) is often rewritten through a Slater determinant which is composed of a lot of single-electron wave functions. This is the so-called Hartree-Fock approximation, where only the exchange effect is considered. In many problems, it is known that the exchange term only is not good. Further developments in order to include the correlation effect into account, many methods, such as the configuration interaction by expanding on many Slater determinants and the quantum Monte Carlo method, etc, have been devised.

Anyway, these approaches are all based upon the wave functions and express the electronic states of solid through the set of wave functions. In the configuration interaction method, the combination of wave functions is very complicated, resulting in severely the limitation of the size of problems.

Meanwhile, for the many-electron problems, another and very different approach called the density functional theory has been proposed. In this approach, the electron density is the quantity, from which the theory is developed. To solve one-electron equations which are derived from the density functional theory is much easier than solving Eq. (1.2). The correlation effect is taken into account, and the size of the system which can be handled is far larger. Since 1980, this method has established a position as one of the main methods of calculating the properties of solid and molecules from the first principles.

The work by Hohenberg and Kohn [1] is now known as a fundamental reference as the density functional theory. In this work, it is shown that the ground states energy of electrons is a unique functional of the electron density. Furthermore, given external potential, It is shown that the ground-state energy can be obtained by minimizing the energy functional, with respect to the electron density. When the density is the true ground-state electron density, this minimizes the energy functional. In a subsequent paper by Kohn and Sham [2], it is shown that the energy functional is recast by using orbitals as $E_{\rm KS}(\{\Psi_i\})$ subjected to the orthogonalization condition of the set of one-electron wave functions $\Psi_i(r)$

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

where $E_{\rm KS}$ is Kohn-Sham functional energy, the i-summation takes over all

one-electron orbits, f_i the number of occupations in i-state, E_{xc} the exchange energy, and $\rho(\mathbf{r})$ is the charge density and given by

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

The wave functions $\Psi_i(\mathbf{r})$ which minimize the Kohn-Sham functional energy in (1.1) satisfy the following eigenvalue equations

$$H_{\mathrm{KS}}\Psi_i = \epsilon_i \Psi_i, \tag{1.5}$$

where H_{KS} is Kohn-Sham's Hamiltonian

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

Here, $V_{\mathrm{H}}(\mathbf{r})$ is Hatree-Fock potential

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

V_{xc} is exchange correlation potential

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

and ϵ_i and Ψ_i denote the eigenvalues and eigenfunctions of the Kohn-Sham equation, respectively.

The wave functions calculated by Eq. (1.5) yield the charge density by Eq. (), which is just $\rho(\mathbf{r})$ appearing in the Hartree-Fock and exchange potential. Hence, the Kohn-Sham equation must be solved self-consistently.

It seems that Eq. (1.6) plays a role of Schrödinger equation of one-electron wave function, but the thought underlying these equation is quite different. For the case of Hatree-Fock, the wave functions are treated as the most important quantity, and the charge density is second one, in other words, a dependent variable. On the other hand, in the density functional theory, the charge density comes first. Wave functions are something expedient, so that they are allowed to vary as far as the charge density is the same.

There are a lot of good reviews about density functional theory and which will be described in the next paragraph ([3, 4, 5]).

1.2 Local Density Approximation

A price of mathematical simplification of the density functional method, which replaces the many-electronic problem by one-electron problem is paid by introducing unknown functional of exchange and correlation $E_{\rm xc}$ of the charge density. Fortunately, there is an easy approximation for $E_{\rm xc}$. The most widely used form of $E_{\rm xc}$ is the so-called local density approximation (LDA). That is, the exchange and correlation energy of uniform electron gas, which is well studied, is used. In this approximation, the exchange-correlation energy at each point of the real space, $E_{\rm xc}(\mathbf{r})$, is assumed to equal to that energy of a uniform electron gas with the same charge density.

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

where t$V_{\rm xc}(\mathbf{r})$ is exchange potential and given by

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

Then $\epsilon_{\rm xc}(\mathbf{r})$ is

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

where $\epsilon_{\rm xc}^{\rm hom}$ is the exchange-correlation energy in a uniform electron gas of that charge density. Actual form of the exchange and correlation energy as the function of the charge density is constructed, based on the most reliable studies about homogeneous electron gas, such as [7] or quantum Monte Carlo method [6]. Within the local density approximation, the exchange and correlation potentials become a local function of the charge density. Tremendous of calculations of for solids and molecules have shown effectiveness and accuracy of this approximation.

1.3 Pseudopotential Approximation [8]

The second approximation which follows the local density approximation is use of pseudopotential. The wave functions of solid is expended here through the set of plane waves. Plane-wave expansion it is uneconomical to describe localized states, such as core states of atoms which exhibit strong oscillations in the core region. Fortunately, the physical and chemical characteristics of many materials are governed by the valence electrons which extend to more wide region, and the core states are insensitive to those properties. We then can make an approximation by using valence electrons solely in describing

the chemical combining characters of materials. Therefore, needed potentials have relatively slowly varying characters and this is desired properties. The wave functions which simulate the valence electrons accords to that are called the pseudo-wave functions. The good reviews of pseudopotential method can be found in Ref. [8].

The pseudopotential are constructed so as they describe as much precisely as possible the electron scattering characters outside the core region. Good pseudo-wave function are called "transferable". In general, pseudopotentials of an atom have different scattering characters in each angular momentum and are non-local. Mathematically, pseudopotentials can be expressed as,

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

where $lm\rangle$ is spherically harmonic functions, l and m are the angular momentum, and the projected angular momentum, respectively. The original bare potential is or course a local potential. Because the true wave function and the pseudo-wave function are matched outside the core region, nonlocality of pseudopotential is limited in the core region. For light atoms, adding of higher than 2 components, $l > 2$, into a set of pseudopotentials is not necessary. Though many empirical potentials had been devised in the past to construct pseudopotentials [8], a great step had been achieved by introducing the concept "norm conservation" of wave function by Hamann[9]. As a result, it is not an exaggeration to say that pseudopotential method came to be used most generally today as a method to solve the problem of solid. In this procedure, a nodeless pseudo-wave function is initially taken so as to match to the true wave function outside core radius r_c. A norm-conserving condition, along with other conditions, the final form of potential is completed. The process is depicted in Fig. 1.1.

In Osaka2k, the Troullier-Martins type by which the efficiency of the calculation is more improved is adopted as pseudopotential by "norm preservation" type[10].

The pseudopotentials defined by expression (1.12) may be called semi-local one. The reason is that it is local for the radical element of the position, though it is non-local for the angle element. Kleinman and Bylander proposed the full-nonlocal type by which the radical element is treated non-local [11]. According to this, the potential is recast by

$$V_{\text{ion}} = V_{\text{L}} + \sum_{lm} \frac{\left|\Phi^0_{lm} \Delta V_l\right\rangle \left\langle \Delta V_l \Phi^0_{lm}\right|}{\left\langle \Phi^0_{lm} \left|\Delta V_l\right| \Phi^0_{lm}\right\rangle} \tag{1.13}$$

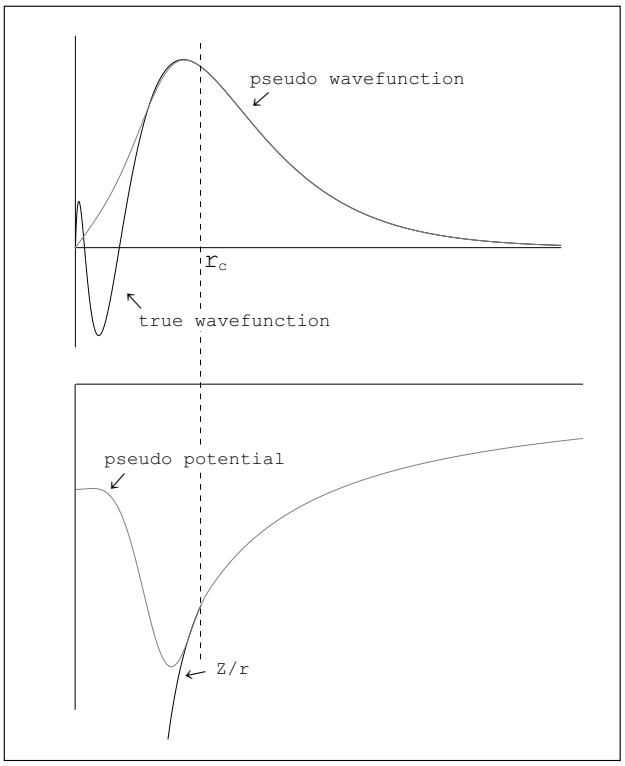

Figure 1.1: **Outline of pseudopotential A true wave function (solid curve) can be replaced with a pseudo-wave function (dashed curve). Wave function and potential of all electrons are the same each other in the chemically important area outside the core radius** r_c

where Φ^0_{lm} is the pseudo-wave function of the atom when the pseudopotential is constructed. ΔV_l is obtained by

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

Giving nonlocal potential in this way, the calculation of non-local part of the potential is greatly accelerated. Further benefit is obtained in calculation of operating of nonlocal potential onto wave functions if the arbitrariness of V_{L} is utilized.

1.4 Plane-wave expansion

There are about $\sim 10^{23}$ atoms in real crystals. It is intractable to solve directly the KS equation for the system with such degrees of freedom, which is virtually infinite dimension. For such a system, it is convenient to use the artificial mathematical tool of Born-von Karman's boundary condition and Bloch theorem[13].

Not only plane-wave method, but almost all the calculation methods of solids undergo a benefit of the periodical boundary condition. For problems of surface of solids, defects, and even for completely disordered solids, we can study these materials by assuming large super cells, which are of course artifacts though.

A mathematical model of a crystal is constructed from three basic translational vectors in the real space:

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

where n_1, n_2, n_3 are arbitrary integers. The lattice constructed from the primitive translational vectors is called Bravais lattice. Crystals are expressed by combining a Bravais lattice and the basis, which is composed of all the atoms in the primitive unit cell.

The reciprocal lattice space is defined for a lattice in the real space. The basic translational vectors of the reciprocal lattice space are defined so as to satisfy

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

where i and j take an integer value from 1 to 3. Now, we have $\mathbf{G}_1$ as

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

$\mathbf{G}_2, \mathbf{G}_3$ can obtained by cyclic change the indeces. The lattice vectors of the reciprocal lattice (reciprocal vector) can be expressed as

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

where n_1, n_2, n_3 are arbitrary integers.

Reciprocal vectors are used for Fourier transformed expression of arbitrary functions with periodicity. When $f(\mathbf{r})$ is a smooth function of $\{\mathbf{R}\}$, we can expand it by

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

Under the periodic boundary condition, the electron state is specified by the wave vector $\mathbf{k}$ and band index n. The wave function has a form of product of a plane wave $\exp(i\mathbf{k}\cdot\mathbf{r})$ and a periodic function $u_{\mathbf{k}}(\mathbf{r})$ with the lattice periodicity,[2]

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

This is what Bloch theorem states, which reduce greatly a problem of solids from nearly infinity to the orders of the number of atoms in the unit cell.

There are many kinds of band calculations, but these are different merely at a point of the ways to express wave functions, i.e., basis functions, except the cellar method.

Because $u_{\mathbf{k}}(\mathbf{r})$ in Eq. (1.20) is a periodical function of the lattice, it can be expanded by the reciprocal lattice vectors according to Eq. (1.19), and then is expressed by,

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

Because plane waves satisfy the Bloch's condition by construction, they provide a good expression as valence electrons in crystal, which are extended over the crystal. Various quantities of the ground states can be expressed neatly by this expansion. On the other hand, a disadvantage of this method is slow convergence (see Ref. [8]).

The total energy Eq. (1.3) is written by plane waves, [14]

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

where $S(\mathbf{G})$ is the structure factor. In addition, the Hellmann-Feynman force and stress, etc, can be evaluated also easily by this plane wave expansion, which will be seen later.

[2] Alternatively, the exponential form can be taken as $\exp(-i\mathbf{k}\cdot\mathbf{r})$. In this program, to keep consistency with TSPACE, we decided to the definition of Eq. (1.20)

The last term of Eq. (1.22) is the so-called Ewald term, which corresponds to the direct Coulomb interaction between ions

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

where κ is the atom index in the lattice cell, and $\gamma_{\kappa,\kappa'}$ is given by

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

Here, $\kappa \neq \kappa'$ is understood in the primed summation. η is so taken as the summation over the real space and that on the reciprocal space in Eq. (1.23a) becomes good convergent. (see. Ref. [15], p. 385)

1.5 Plane-wave cutoff

Eq. (1.21) is a summation over infinite number of reciprocal lattice vectors $\mathbf{G}$. In actual calculations, this summation of course has to be truncated at somewhere. In practice, only plane waves whose kinetic energy is smaller than the certain cutoff energy E_{cut} (atomic unit) are chosen, as

$$|\mathbf{k}+\mathbf{G}|^2 < E_{\mathrm{cut}}. \tag{1.24}$$

This means that a sphere is defined in the reciprocal lattice space, and all the wave vectors inside the sphere (its radius k_{c}), are included. See Fig. 1.2. Then, the plane wave cutoff radius is given by

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

Number N_{pw} of plane waves inside the cutoff radius k can be estimated by

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

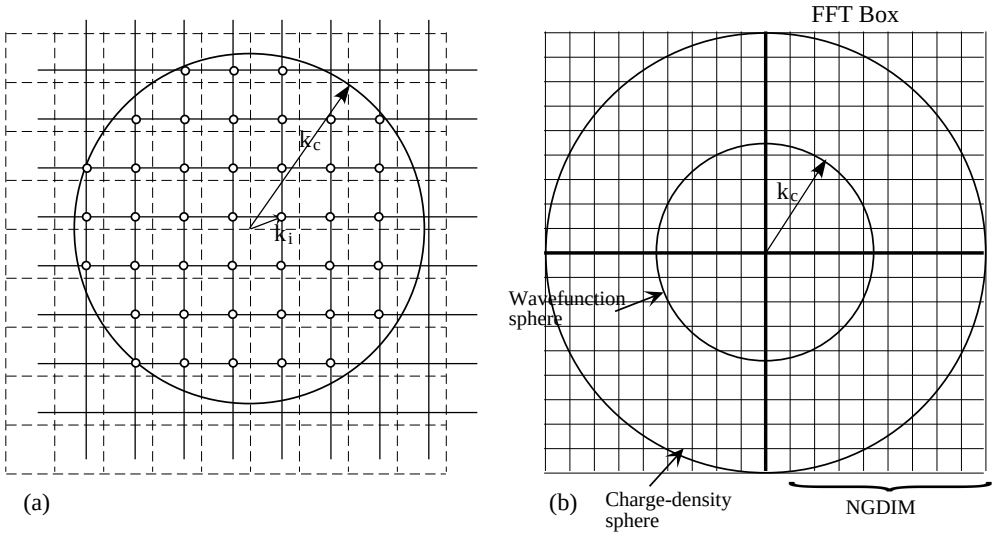

Figure 1.2: (a) Expanding plane waves for $\mathbf{k}_i$. (b) relationships between plane wave sphere, the charge density sphere, and FFT box.

where Ω_c is the volume of the primitive lattice cell.

Number $N_{\rm pw}$ of plane waves needed for good convergence depends on the atom type in the primitive unit cell. In general, the systems containing elements of 1st row in the periodic table need a cutoff energy larger than system containing only elements of 2nd or 3rd rows.

1.6 k-space summation

In the calculation of the total energy and forces, taking an average of operator $\hat{Q}$ which acts on the Kohn-Sham wave functions is often needed.

According to Bloch's theorem, the average over all the wave functions is calculated by

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

where $Q_{\mathbf{k}}$ is given by

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

For example, in the case in which $\hat{Q}$ is Kohn-Sham Hamiltonian, $Q_{\mathbf{k}n}$ is the eigenvalue of Kohn-Sham equation, and Q is the average value of the band energy of n-th band.

In the numerical calculation of Eq. (1.27), the integral can be replaced by the summation of finite points in k-space, and it usually takes a lot of time. However, with choosing wisely enough a set of special points, we could obtain as accurate as by using much more k points. This is called the special point sampling method. In our program, the method by Monkhorst and Pack [16] is used as the standard special point sampling method. The mesh of k points is created by three integers of N_1, N_2, and N_3. These integers determine the density of k points in a primitive unit cell of reciprocal lattice. A general point of the mesh is given by

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

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

where p takes a value from 1 to N_i. This mesh makes $N_1N_2N_3$ pieces of k points in the Brillouin zone. Therefore, the integral of (1.27) is replaced with the summation of discrete k points.

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

Some k points are occasionally points on symmetry line or planes. If some k points are connected each other by symmetry, it is enough to solve the Kohn-Sham equation at only one point of the group of symmetry-connected points (stars). By using symmetry, computation time can be greatly saved. In Osaka2k, the number of k points is decreased as much as possible by properties of crystal symmetry and time-reversal symmetry.

The quality of the special-point sampling is accessed by the cutoff vectors of the real space. Periodical functions of the reciprocal space like $Q_{\mathbf{k}}$ can be expressed by the Fourier series in the real space

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

where $B_{\mathbf{R}}$ are expansion coefficients. B_0 is the average of $Q_{\mathbf{k}n}$ over the Brillouin zone. Usually, the Fourier coefficient $B_{\mathbf{R}}$ decreases rapidly with increasing $|\mathbf{R}|$. Inserting (1.32) into (1.31), we get

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

where $\phi_{\mathbf{R}}$ can be written by

$$\phi_{\mathbf{R}} = \frac{1}{N_1N_2N_3}\sum_{rst} e^{i\mathbf{k}_{rst}\cdot\mathbf{R}}. \tag{1.33a}$$

The summation of $\phi_{\mathbf{R}}$ over $\mathbf{R} \neq 0$ can be regarded as an error of Q for a particular set $\{rst\}$ of sampling points. By classifying $\mathbf{R}$ through $|\mathbf{R}|$, *i.e.*, shell structure of $\mathbf{R}$, the summation of Eq. (1.33) can be grouped by shells, as

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

As the shell sum vanishes to further extent, the qualify of the sampling points becomes better.

1.7 Conjugate gradient method in Kohn-Sham functional

Next, it is necessary to solve Eq. (1.3). A traditional method to solve this is to diagonalize Eq. (1.5) self-consistently so that the output charge density and input charge density become to match. In this method, the output charge density is returned to as the input of the next step, by mixing it with the old input charge density. How set the mixing rate is subtle problem. Especially when the size of a crystal becomes large, adjustment of the mixing rate is indeed a tough business. The conjugate-gradient method in Kohn-Sham functional is an excellent method to solve this difficulty. We will explain in detail below.

Based on the arguments so far made, to seek the ground states of a crystal is equivalent to a problem of minimizing the total energy $E_{\mathrm{KS}}(\{c_{\mathbf{k}+\mathbf{G},n}\})$ by varying plane-wave expansion coefficients $\{c_{\mathbf{k}+\mathbf{G},n}\}$ of all the occupied bands.

It is known that conjugate-gradient method is an efficient mathematical technique to find the minimum point of multi-dimensional functions $f(\mathbf{x})$. Especially, when $f(\mathbf{x})$ can be approximated by quadric form,

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

and when $\mathbf{A}$ is positive-definitive, one can always find the minimum point, at least in principle.

Because a gradient of function $f(\mathbf{x})$ at $\mathbf{x}$ is given by

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

the minimum point $\mathbf{x}$ satisfies the following linear equation

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

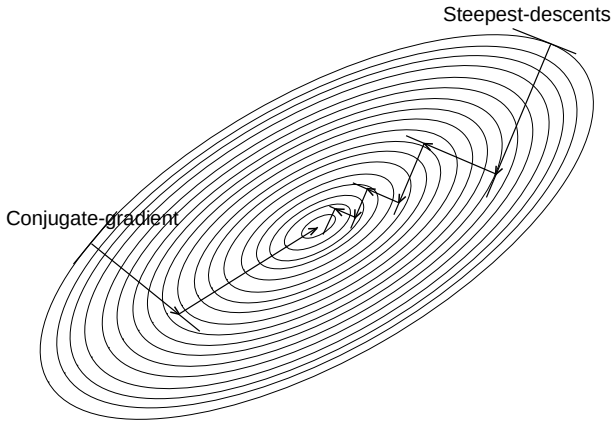

Figure 1.3: Comparison between the steepest descent method and conjugate gradient method

A primitive approach to solve this minimization problem is first to evaluate Eq. (1.35) at a trial point $\mathbf{x}_0$, which is followed by the line-minimization in the gradient direction. When the minimum point $\mathbf{x}_1$ on the searching line is found, the gradient of that point is evaluated and then the minimization in this new gradient direction is performed again. This process is repeated until some criteria is met. This method is called steepest-descent method. Though this is an intuitively easy way, the process is inefficient, as shown in Fig. 1.3: there are many steps repeated in the valley of potential.

When this inefficiency of the steepest-descent method is analyzed, one can find that the same searching direction is examined by many times. Once one direction is searched, further repeating is waste of time. In the conjugate-gradient method, this unnecessary repeating is avoided by a clever way.

In the conjugate-gradient method, it is assumed that the minimum points $\mathbf{x}_i$ were found in the searching direction $\mathbf{h}_i$ in i-th step. Next, the gradient $\mathbf{g}_i$ is evaluated at this point. With taking $\mathbf{x}_i$ as the new starting point, the searching direction $\mathbf{h}_{i+1}$ of the next step $i+1$ is given by

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

where the mixing ratio γ_i is determined by

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

This searching direction is called the conjugate-gradient direction. As can be seen in Fig. 1.3, by searching in this direction the minimum point is found very efficiently.

Moreover, in this method, the explicit form of matrix $\mathbf{A}$ is not required, because $\mathbf{A}$ always appears in the form of the product with vectors $\mathbf{x}$. Therefore, if $\mathbf{A} \cdot \mathbf{x}$ can be efficiently evaluated, then not only the calculation becomes fast, but also saves the memory.

According to the density functional theory, the energy functional has the minimum at the true ground-state charge density, while it is increased for all the other density distributions. Therefore, the conjugate-gradient method discussed above is suitable to find the ground state of a crystal. However, in adapting this method to the present problem, there are troublesome restrictions that wave functions must be normalized and that wave functions must be orthogonal each other. This makes the conjugate-gradient process in the present problem more complicated.

These difficulties were overcome, and a more effective conjugate-gradient method was proposed by Teter, Payne, and Allan(TPA) [18, 19]. Osaka2k follows the algorithm of this TPA algorithm.

In this algorithm, the set of initial wave functions that are orthonormalized are prepared for the occupied bands of each k point. They are generated from random numbers by default. The conjugate-gradient minimization is applied to a single band of the first k point in order to minimize its contribution to the total energy, while the other bands are fixed. Conjugate-gradient processes are performed several times to the first band, and then turn to the next band. After scanning all the bands of the k point, proceed with the next k point. Scanning all the k point constitutes one iteration. This whole process is also repeated several times.

The conjugate-gradient process to one band of n-th is described in more detail below through the referring to figure 1.4.

1. Kohn-Sham energy functional E_{KS} corresponding to the wave function $\{c_{\mathbf{k}+\mathbf{G},n}\} = \mathbf{C}_n$ of the n-band leads to one electron Hamiltonian H of Eq. (1.6). The expected value corresponding to the wave function $\mathbf{C}_n^m$ of n-th band in the m-th step is given by

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

   As a result, the residual error $\mathbf{R}^m$ to this Hamiltonian determines the direction of the steepest-gradient direction,

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

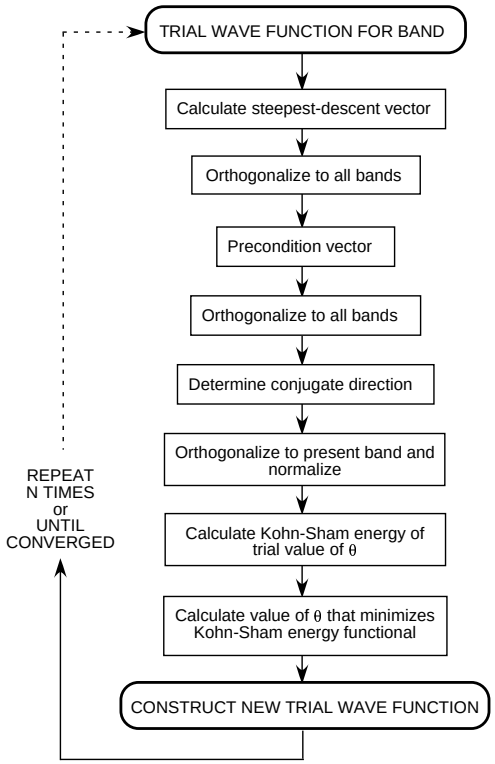

Figure 1.4: Flow diagram of the direct minimization method

The mean square of this $\mathbf{R}^m$ is called the residual error ξ^m of wave functions.

2. This residual error vector $\mathbf{R}^m$ is corrected so as to orthogonal to other bands of the k point:

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

3. To accelerate the convergence, this corrected residual error vector $\mathbf{R}'^m$ is preprocessed (preconditioning). That is, a suitable weighting matrix $\mathbf{K}$ is multiplied in order for all the diagonal elements of $\mathbf{R}'^m$ to be of the same order of magnitude,

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

In practice, the matrix $\mathbf{K}$ is so constructed as a diagonal matrix, whose diagonal element is given by the inverse of the corresponding element of $\mathbf{R}'^m$ roughly.

4. Modifying $\mathbf{R}''^m$ again so that it is orthogonal to all the bands including itself.

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

5. Using this $\mathbf{G}^m$, the conjugate gradient is obtained. That is,

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

is given for the conjugate gradient. The mixing ratio γ^m can be obtained from $\mathbf{G}^m$ according to Eq. (1.38). γ^0 can be set to be 0 as the initial condition.

6. $\mathbf{F}^m$ is orthonormalized with the present band again. We denote it by $\mathbf{D}^m$. By this way, the line minimization problem in the direction $\mathbf{D}^m$ becomes in turn the minimization problem of the total energy $E_{\mathrm{KS}}(\theta)$ as a function of θ.

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

7. Since $E_{\mathrm{KS}}(\theta)$ is a function of the density, and the density is proportional to the square of the wave function, if $E_{\mathrm{KS}}(\theta)$ can be approximated to be linear in the density, $E_{\mathrm{KS}}(\theta)$ may be expressed by

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

Because there are three unknown numbers in Eq. (1.46), three equations are necessary. Value $E_{\mathrm{KS}}(0)$ with $\theta = 0$ has already been evaluated, and its derivative is also obtained easily as,

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

Because $\mathbf{H} \cdot \mathbf{C}_n^m$ has already been evaluated, Eq. (1.47) is obtained by only taking the scalar product of vectors. B in Eq. (1.46) is half a value given by Eq. (1.47).

8. The remaining variable can be obtained either by evaluating $E_{\rm KS}(\theta)$ with $\theta_1 \neq 0$ or by calculating the second derivative of $E_{\rm KS}(\theta)$ at $\theta = 0$. We will use the later way here. By this way, A is given by $A = (1/4)\partial^2 E_{\rm KS}/\partial\theta^2$. Then, $\theta_{\rm min}$ minimizing Eq. (1.46) is determined by

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

1.8 Fast Fourier transform

An important part of calculation in the conjugate-gradient method is to operate the Kohn-Sham Hamiltonian on the wave functions. In the program, this step is carried out by using the Fast Fourier Transform (FFT). Osaka2k utilizes the fact that the kinetic energy is local in the reciprocal space, and that local potential is diagonal in the real space.

Given the number of basis function $N_{\rm pw}$, calculation of the kinetic energy from in the reciprocal space needs arithmetic operations $\mathcal{O}(N_{\rm pw})$ in Eq. (1.22). On the other hand, because the local potential is a matrix of the size $N_{\rm pw}$, the calculation in the reciprocal lattice space needs operations of $\mathcal{O}(N_{\rm pw}^2)$. This operation can be accelerated by rewriting with a real space representation,

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

From Eqs. (1.20, 1.21), we see that summing $c_{\mathbf{k}}d + \mathbf{Gd}$ gives

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

Therefore, the evaluation of Eq. (1.49) is:

1. To convert the wave function $c_{\mathbf{k}+\mathbf{G},n}$ expanded in the reciprocal space into the representation $u_{\mathbf{k}n}(\mathbf{r})$ in the real space.

2. To take the product of $u_{\mathbf{k}n}(\mathbf{r})$ and $V_{\rm L}(\mathbf{r})$ in the real space.

3. To convert the result to its representation in the reciprocal space.

Though this three-step calculation seems more involved than the direct product of $\mathbf{V}\cdot\mathbf{C}$ in the reciprocal space, it actually is superior in terms of computation speed.

Because the operation of 2 is actually a diagonal sum in the real space, the number of floating-point arithmetic is only $\mathcal{O}(N_{\rm pw})$. Operations of 1 and 3 needs only the number of floating-point arithmetic of order $\mathcal{O}(N_{\rm pw}\log(N_{\rm pw}))$, when FFT is used. The number of overall operation is therefore still order of $\mathcal{O}(N_{\rm pw}\log(N_{\rm pw}))$. This technique was devised by Car and Parrinello[20] and has become widespread.

It may be worth to notice the region on which FFT is performed. The wave functions are expressed in a spherical region with the radius $k_{\rm c}$, in which $N_{\rm pw}$ plane waves are involved, as shown in Fig. 1.2 (a). On the other hand, in order to express the charge density faithfully for varying wave functions, we must take a sphere of the twice radius in the reciprocal space. See the figure (b). As a result, the rectangular region on which FFT is performed is the length of$4k_{\rm c}$. The mesh in the reciprocal space is almost the same as that in the real space. According to Eq. (1.26), the number of mesh points in the real space, $N_{\rm FFT}$ is given by

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

1.9 Hellmann-Faynman forces and stresses

Hellmann-Faynman theorem enables us to calculate atomic forces and stresses easily.

In a material, a force exerted on an atom whose position is given by $\mathbf{R}_I$ is given by energy derivative with respect to $\mathbf{R}_I$

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

By this definition, in order to estimate atomic forces, the total energy must be evaluated at several (at least two) points of $\mathbf{R}_I$.

Hellmann Faynman theorem reduces this task considerably,[21] In vertue of this theorem, Eq. (1.52) can be recast as

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

At a glance, difference between Eq. (1.52 and (1.53) might seem not significant. But, the difference is definite. In Eq. (1.52), wave function Ψ has implicitly dependence on atom positions. Once atom positions are changed, Ψ must be recalculated. On the other hand, in Eq. (1.53), Ψ is the wave function at the equilibrium positions. Schrödinger equation is once

solved, and only once is enough. What you have to do is merely integration of wave function over new external potential. The latter task is much easier than performing another SCF calculation.

Evaluation of forces and stresses can be done straightforwardly in plane-wave expansion. Because the basis functions are fixed in the real space, application of Hellmann-Feyman theorem is simple and yield no correction term, such as Purley correction.

For concrete expressions, see [14] for atom forces, and se [22, 23] for stresses.[3]

Information of atom force is utilized for optimization of atom positions, Similarly, information of stresses is utilized for cell optimization.

It should be noted that in general convergence speed about force and stress is much slower than that of the energy. The total energy is a variable with respect to the change in wave functions. This means that the variation of the total energy around the equilibrium state is second order in the variation of wave function. On the other hand, the variation of force is only the first order.

[3] In Ref. [23], in Eq. (2), the sign appearing in front of kinetic energy is mistyped as $-$.

Chapter 2

Load Map

2.1 Overview

A electronic-structure calculation package of "Osaka2002" based on pseudopotential method is actually a set of several program codes. The relationship between them is shown in Fig. 2.1.

In brief, given a crystal, which is constructed by `cryst`, prepare atomic potentials of constituting the crystal by `atom`, carry out self-consistent field (SCF) calculation by `pwm`. `inip` does preparation to pass some parameters to `pwm`.

The core of Osaka2002 is `pwm`. Once SCF calculation is completed, we have several choices to proceed further. Among them are optimization of the crystal, phonon calculation, molecular dynamics simulations.

If you need to know details of electronic spectra, band and DOS calculations can be obtained by `pwbcd`. `pwbcd` has an option to display wavefunctions in the real space.

`atom` is a program to generate pseudopotentials, and is independent of the rest of the package Osaka2k.

Throughout all the calculations, full use of crystal symmetry was made. This is accomplished thanks to a library `TSPACE`.

2.2 Preparation

directory structure

After downloading the source codes, expand in a suitable directory. On expand, you will see four directories, as shown in Fig. 2.2. Users should create

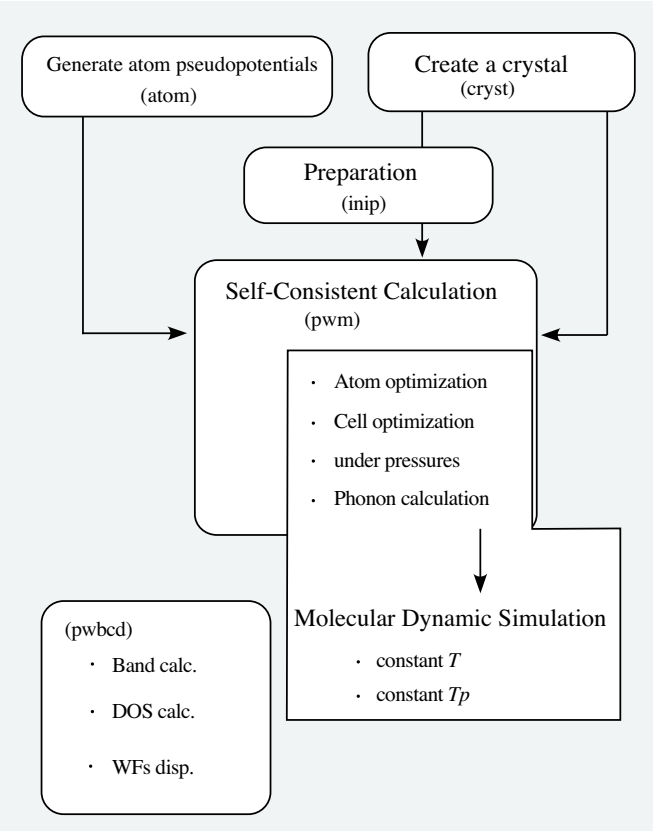

Figure 2.1: Overiew of Osaka2002

a directory `/bin`, in which executable codes are stored. For users convenience, we add a directory `/drivers`, which is not shown in the figure. In this directory, machine-dependent files and Makefile are stored. Users should add a working directory `/data`, in which you will carry out calculation.

All the source codes are put in directory `/fsrcs`, along with Makefile. These are copied into directory `/bin`, and are compiled to create executable codes. These executable codes will be copied or linked into a working directory which contains a crystal data you want to calculate, such as `/sidat` in the figure.

Directory `/input` contains template files for input.

The current version of Osaka2002 is featured by being reusable; once

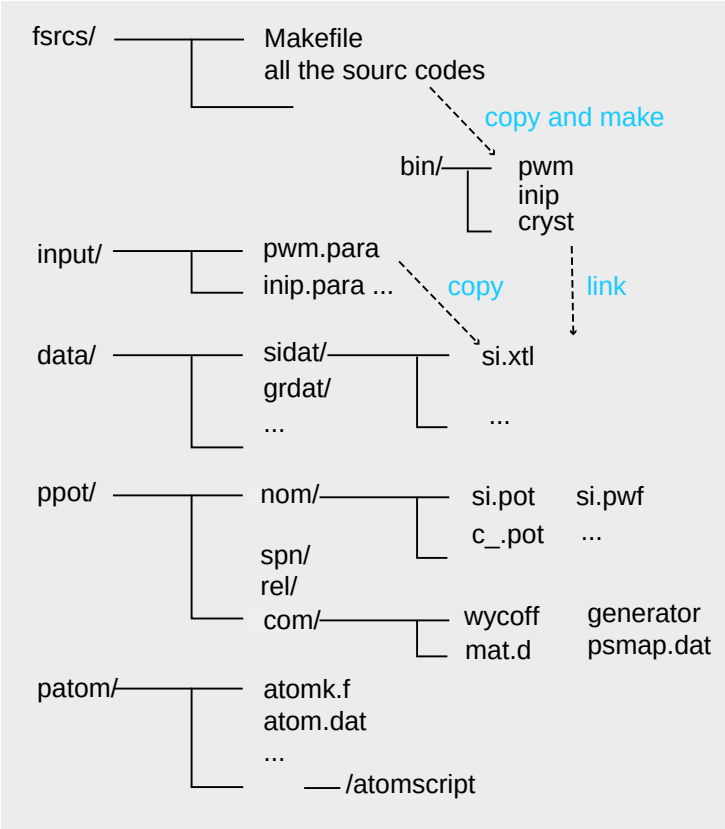

Figure 2.2: Directory structure of Osaka2002

executable codes are compiled, there is no need to recompile them when a crystal or calculation conditions are changed. In a previous version written by old Fortran, many calculational parameter should be given before compilation, and thereby recompilation was needed whenever these parameters or crystals were changed. Only one exception is `TSPACE`, because this was written at so old age.

compile

First of all, find `Makefile`, `drivs.f90` suitable to your machine in `/drivers`, put these in `/fsrcs`. Then, edit `/drivs.f90` as described latter. When you like to get benefits of the maximum performance of your machine, further arrangements, such as in `fft_util.f90`, may be needed. Some of techniques

are described in "Installation Manual of Osaka2002".

Copy Makefile in `/bin`, type

```
% make prep
```

Then, all the needed files are copied in `/bin`.

All the executable codes are created by Makefile. For `inip`,

```
% make inip
```

For `pwm`,

```
% make
```

After comilation, unnecessary files will be deleted by commands

```
% make clean
```

or

```
% make clobber
```

For `pwbcd`, which carrys out band, DOS calculations along with others,

```
% make pwbcd
```

Atom potentials are created in `/patom`, the data files should be restored in directory `/ppot`.

2.2.1 machine dependence

After copied necessary files in `/fsrcs`, modify these in order to meet your michine. In `/drivers`, machine-dependent codes such as `Makefile`, `drivs.f90`, are put in according to specific machines. Suitable ones should be moved to `/fsrcs`, and add modification.

In `drivs.f90`, find a subroutine `potdir`, and edit a part

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

in order to meet your environment. This is the directory in which atomic potentials are restored. Note that `LEN` is exactly matched to the length of `datadir`. It is our observation that, even though `datadir` is correct, the length is often wrong, so that it failed to find the potential files.

`pwm` uses a mathematical library `lapack`. Describe the method to link this library in Makefile, according to your system construction. `TSPACE` is troublesome, because this is written by old style Fortran. Some of tips of handling `TSPACE` is described in Appendix of "Installation Manual of Osaka2002"

> For TSPACE users
> `TSPACE` enclosed in this package is basically the same as that of Ref. [24]. However, in order to make more flexible to treat variety of materials, some parameters are written in a separate file `TSPARAM` and the file is included at the compilation time. Therefore, `TSPACE` in this package must be used.

2.3 Input parameters

Every program in the package requires input data, as usual. These data are commonly written in a file named as `*.para`. The format of these files will be described in order to appear. Here, notices applied commonly in these files are described.

One of problems which came to be clear on the course of the program development, is that, at every time program revision is made, input parameters are subjected to change. In case this is often happened, to rewrite the manual only due to minor changes is indeed troublesome for both of the writer and users. In many cases, actually such a minor change of parameter does not relate to the uses of average users. On the basis of this painful experiences, we decided that a standard file `*.para` explicitly keeps only few parameters, which are frequently used. Others are added when needed. The way to add these parameters is to use option input.

2.3.1 Input options

Options are to be added according to user's special needs. Because these extended features are continuously progressed as the program is developed, it is not suitable to describe this kind of text. These things belongs to details

of revision, so that these are described separately in a form of Technical Report Series associated with Osaka2002.

Here, we describe only general rules in adding options, which are common in every `*.para` file.

First of all, keep distinction of upper and lower cases, i.e., options are case sensitive.

After describing general input parameters, make one empty line, then type

```
OPTION BEGIN
```

specify each option followed by

```
OPTION END
```

During these description, do not insert blank lines.

There are two types of options.

1. toggle variables

   ```
   fermi_broadening ON
   ```

   Type the name of variable, and put one (and only one) space, followd by `ON/OFF`.

2. value variables

   ```
   pressure=
     1.5
   ```

   Type the name of variable, followed by= without any space. Then, after carriage return, its value is typed after at least one space. When a real number is entered, type for example `300.` even when the value is just 300.

The available options are subjected to change as the program code is developed, refer Technical Report Series, or a source file `auxinp.f90`. In directory `input`, there is a file `opts_list`, which lists all the implemented options.

This completes file preparation. Now, we can proceed to creation of atomic potentials in the next chapter

Chapter 3

Atomic pseudopotentials

One feature of contemporary pseudopotential method is that a part of creating potential can be separated from the rest of bulk electronic structure calculation. Currently, there are several program codes to generate pseudopotentials. Package Osaka2k uses one of such program, `atom`, which is developed by Troullier and Martins. Hence, we cannot owe responsibility for this program, even though we will explain sometimes principles underlying this code. Although utility of this program is really wide, we restricted our concerns to those related to `pwm`.

The source code is `atomk.f` in directory `/patom`.[1]。

To create `atom`, type

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

After creating an executable `atom`, generate atom pseudopotentials needed one by one [2]

3.1 input file

First, edit an input file `atom.dat`. The format is as follows,

The meaning of each line is as follows

1st line type of calculation (*itype*) and title(*ititle*).

As types of calculation,

itype=

[1]The original code is `atom.f`. Only change made in `atomk.f` is output format in order to meet the input of `pwm`

[2]Although the code is well developed, there is still some machine dependence. These are referred to "Installation manual"

Table 3.1: Format of `atom.dat`

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

- **ae:** all-electron calculation
- **pg:** generation of pseudopotential
- **pe:** generation of pseudopotential with core correction with respect to the exchange-correlation term
- **ph:** generation of pseudopotential with core correction including the Hartree term
- **pt:** test of pseudopotential
- **pm:** test of pseudopotential and modification of valence electrons

2nd line kind of pseudopotential (*ikerk*)
ikerk=

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

3rd line atom name(*nameat*) and options(*icorr* + *ispp*).
nameat is just atom name. *icorr*+*ispp* is consisted of three characters. The first two corresponds to *icorr* and gives the type of electron-correlation functional. Those available are
icorr=

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

The third character corresponds to *ispp* and gives a selection *ispp*=

r relativistic calculation

s spin polarization

␣ none

4th line charge number of nucleus ($znuc$), charge of inner core (zsh), core radius (rsh), maximum radius($rmax$), parameters of radial mesh(aa, bb).

5th line number of core orbitals(n_{core}) and number of valence orbitals(n_{val})

after 5th line occupation of the valence orbitals specified by n and $l(zo)$. The orbital energy as option input (evd).

last line cutoff radii for generating pseudooptentials(r_c).
Further parameters are added when core correction is needed ($cfac$ and $rcfac$).

Atomic orbitals are specified by three quantun numbers, i.e., the principal quantum number n, angular-momentum l, and its azimuthal component m, even when spin freedom can be ignored. In `atom` (and most of other programs of this kind), spherical potentials are assumed. This treatment means, for example, when there is only one electron available among p_x, p_y, and p_z orbitals, each orbital is occupied equally by 1/3 electron. This keeps the dimensionality of problem within only one dimension. Otherwise, the problem would become essentially three dimensional one, for which solutions become much more complicatd. Therefore, third quantum number m is ignored.

In the following, we explain how to specify these parameters, taking Si as example.

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

The first line gives the code of calculation kind, name of atom, while the second gives the kind of pseudopotential. For now, we write as

pg generation of pseudopotential

tm2 Improved Troullier and Martins

The third line is written as shown. The forth line is ignored. Leave it alone.

The fifth line gives the numbers of core and valence orbitals, which are specified by two quantum numbers n and l. In the present example of Si, the neutral atom is configured by $(1s)^2(2s)^2(2p)^6(3s)^2(3p)^2$. This means that there are three core orbitals; $1s$, $2s$, and $2p$, while two $3s$ aned $3p$ orbitals are included as the valence orbitals.

Next, occupation numbers are followed. In each line, one valence orbital is described by specifying n, l, the number of occupation of down-spin and up-spin. In the present case, spin freedom is ignored, the occupation number of only down-spin is used. For Si, two electrons are occupied for $3s$ and two are occupied $3p$ orbital.

The last line gives the cutoff radii for generating pseudopotentials in order of s, p, d, f orbitals. Values are given in atomic unit.

3.2 execution

We explain the basic usage of `atom`. Advanced usage of it should be referred in related Technical Reports.

1 All electron calculation First, all electron calculations including all core orbitals must be performed. Set `pg` as the type of calculation, [3]

[3] All electron calculation is originally performed by setting `ae`. But, in the process of generating pseudopotentials, all electron calculation is performed before creation of pseudopotential.

At this moment, you may do not wary about the values of the cutoff radii. Putting a data file `atom.dat` in the same directory as `atomk`, execute `atomk`.

2 Read an output file `atom.out` In an output file `atom.out`, results of all-electron calculation is written. The eigen-energy of all the orbitals are listed, along with information about the shape of orbitals along in the radial axis. Users are encouraged to plot orbitals. For doing so, `fort.11` can be used.

In file `atom.out`, after

```
radial grid parameters
```

the characteristics of each orbital in the radial direction is listed, as

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

This describes the features of $2s$ orbital. This orbital has two extreme and one zero point other than the origin. The position of zeros are given by `r zero` in Bohr units. `r extr` indicates the position of extreme. Users should check if the obtained orbitals behave as expected.

Next, determine the cutoff radii for generating pseudopotentials, by seeing the shape of valence orbitals. There is to a large extent arbitrariness in choosing the cutoff radius r_c. This arbitrariness causes beginners confusion (not only beginners but experts too). At the first trial, it may be taken to be 1.1 $\sim$ 1.6 times larger than the position of the most outer extremum, by rule of thumb. It can be set to be inside of the most outer exremum, as far as it still is outside of the most outer zero point. In general, as r_c is set to be larger, the obtained potential becomes soft and consequently good convergence is achieved. A bad news is degradation of transferability.

One feature of the Troullier-Martins type is that r_c can be larger than others such as Bachelet et. al. [12]. The cutoff radii are written in the last line of `atom.dat`.

3 Generating pseudopotentials After deleting all the previous output files, again execute `atom`. Then, all the needed files are created. The pseu-

dopotential and pseudo-wave functions are written in `fort.10`. Users are again encouraged to see these as graphics output.

In `fort.10`, wavefunctions and potentials are listed in order s, p, ... Each data are listed as two column data; the first is the radial position and the second is the value of wavefunction/potential. At the first, the number of mesh points appears.

The pseudopotential data as output is the bare-ion one, which the electronic contribution has been removed, and the wavefunction $u(r)$ is normalized as it were. This means that it holds

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

but not $\int \cdots r dr$.

4 Give the names for data files Finally, take binary files `pseudo.dat01` and `fort.13` among various output files. The former is the potential data, while the latter is the pseudo-wave function data. These files are actually used in the following calculations. Rename these and saved in directory `/ppot/nom`. The file name of potential should be (atom name).pot, and the name of wavefunction should be (atom name).pwf. Atom name is given by two characters. In the present case, these are `si.pot` and `si.pwf`, respectively. In case of C, underscore should be followed, such as `c_.pot` and `c_.pot`. The name characters should be written by lowercase. It should be noted that `*.pot` and `*.pwf` files are usable only on the current platform, since these data are written binary format. When platform is changed, then potential and wavefunction data should be recalculated on that platform.

5 Repeat if necessary The above process is repeated for used atoms.

To what extent can we take as the valence electron?

As studying various materials, you will find it problem to determine which electron is counted as the valence electron and which is counted as the core electron. For light elements, such as Si, this seems definite. But, for $4f$ elements and heavy elements, such distinction becomes difficult. In addition, we are further embarrassed with questions as to what configuration of electrons should be used as the reference state when pseudopotentials are constructed.

All of these problems are essential problems in the pseudopotential method. As a guide, we supply input templates for all the elements in `/patom/atomscripts`.

However, this is by no means the best set, nor even correct. When you begin to suspect these supplied data, and when try to change the input data to see how the result is influenced, then this is a sign that you are qualified as a real and active researcher in this field. You will search the literature and discuss a problem to other researchers.

Chapter 4

Construction of crystals

4.1 Input of crystal data

Crystals are described by the lattice of the unit cell and the atoms constituting the lattice. The *primitive unit cell* is the minimum lattice which preserving the translational symmetry. However, more large cells called as *conventional unit cell* are often used due to practical reasons.

In the primitive unit cell, the unit translational vectors are expressed by a set of $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$, while in the conventional unit cell the unit translational vectors are expressed by a set of $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$.

When referring the coordinate (h, k, l) without any notice, we cannot say which of $h\mathbf{R}_1 + k\mathbf{R}_2 + l\mathbf{R}_3$ or $h\mathbf{a} + k\mathbf{b} + l\mathbf{c}$ is meant by. The former is said as the expression in the primitive base, while the expression in the conventional base. Correspondingly, the reciprocal lattice is expressed by $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$ in the primitive base, while $\mathbf{a}^*$, $\mathbf{b}^*$, $\mathbf{c}^*$ in the conventional base.

Usually in the crystallographic literature, a crystal is described in the conventional base, by using three lengths, (a b c), of the lattice vectors and its three angles (α, β, γ), where α means the angle between b and c axes, and so on.

In `cryst`, by using a, b, c, α, β, and γ, the lattice vectors are translated in the cartesian coordinated, as

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

where r is given by

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

In case of the rhombohedral crystal system, definition of the cartesian coordinates given by Eq. (4.1) is inconvenient, because the 3-fold axis is not in the z-axis. Therefore, only in this case the 3-fold axis is taken as in the z-axis, as an exceptional case. The cartesian coordinates of this case is shown in Fig. 4.1. The 2-fold axis is taken as in the x direction.

Basically, the physical properties calculated are independent of the way of the cartesian coordinates. But, for example, the eigenvectors of phonons, elastic constants are expressed only when the coordinate system is specified.

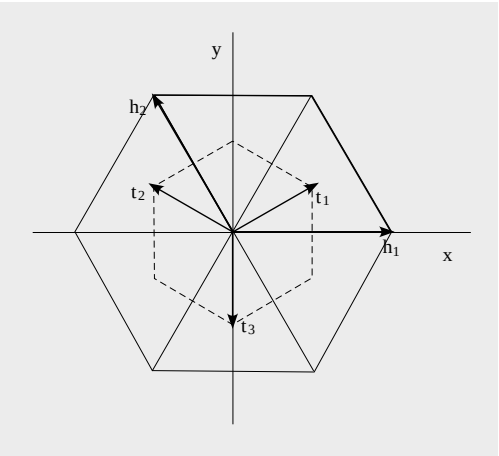

Figure 4.1: Way of taking the crystal axis for the rhombohedral crystal system. The 2-fold axis is taken in the x direction, while the mirror plane intersects to the y axis.

When you try to read the source code, you will find that conversion of the coordinates is scattered over the codes. `TSPACE` employs the conventional base, while `pwm` employs the primitive base. In addition, for the above reason, the cartesian coordinates are sometimes used. Therefore, conversions between different coordinate systems are inserted indeed many times, which makes the code more difficult to be read.

The crystal data are written in a file `name.xtl` For example, crystal GaAs is described by `gaas.xtl`,

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

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

This data format accords to that of a commercial program "INSIGHT II" of MSI. Therefore, it can be read by "INSIGHT II".

In the data form, first `TITLE` and `DIMENSION 3` appear. Then the lattice parameters follow. These are expressed as

a, b, c, α, β, γ

in the conventional base in the Å or degree units.

Next comes the number of the space group and `QUALIFIIER` if desired.

Then, the relative coordinates of each atom is listed. Note that the coordinates are expressed by being relative to the conventional unit. You will find this treatment convenient in most cases. However, there is many confusions in the trigonal system. In this case, we accords to the policy described in Ref. [26], where the problem is well analyzed. When the Bravais lattice belongs to the rhombohedral unit, the relative coordinates should be described with respect to the rhombohedral axes. In this case, `QUALIFIER` is set to be `Rhombohedral`. However, even when the Bravais lattice belongs to the rhombohedral unit, the relative coordinates are sometimes described with respect to the hexagonal unit. In this case, these coodinates can be used as it were by setting `QUALIFIER` to be `Hexagonal`.

Description of crystals is indeed complicated. As to further reading about this problem along with others, refer to a supplement **??**.

Users are required to distinguish several quantities concerned with the number of atoms in the unit cell. [1]

$N_{\rm at}$ **:** the total number of atoms in the primitive unit cell

[1] In the current version of Osaka2k, all the include files in `pwm` were discarded, so that these parameters are no longer used explicitly. Despite, it is still important to know the meaning of these parameters. Actually, even in the current version, `TSPACE` still uses its include file, in which some of these parameters appear.

$N_{\rm spe}$**:** the number of chemical elements constituting the crystal

$N_{\rm ka}$ **:** the number of distinct atom sites (irreducible sites), which are not connected by symmetry

Distinction between $N_{\rm spe}$ and $N_{\rm ka}$ is important. In the former classification, all the atoms are counted as one, if these atoms are the same chemical element. On the other hand, in the latter classification, even when the atoms are the same chemical element, they are counted as different kinds if they cannot be transformed by crystal symmetry.

For example, in graphite crystal, these parameters are

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

In a file `*.xtl`, it is not necessary to list all the atom coordinates in the primitive unit cell. Listing only prototype atoms at the irreducible sites suffices. Hence, only number $N_{\rm ka}$ of atoms appear in the list. Other atoms are created by symmetry operations in `cryst`. In the package or home page, files many crystals are available, so that you can exercise how to describe crystals.

In a line, `NAME` comes first. This name is arbitrary. Next, three relative coordinates come, followed by the name of potential. This name of potential is the one by which the correct potential data file is searched, so that it must be the same as the name of the potential. For example, for Si atom, you should type `si`. For C, you shoud type one character `c` followed by one space, as `c`␣.

The remaining data are not used, accordingly you can omit them. [2]

Description of `QUALIFIER` accords to "International Tables for Crystallography" (ITC) [25]. These available descriptions are

`ORIGIN_1` or `ORIGIN_2`

For the trigonal system,

`RHOMBOHEDRAL` or `HEXAGONAL`

For low-symmetry crytals,

`UNIQUE_b` or UNIQUE_c

[2] In a previous version, the next data of valence number was required. Now, this is not necessary. These data are read in from `/ppot/com/psmap.dat`

and

UNIQUE_b,CELL_1, 2 or 3

Due to the restriction of `TSPACE`, for base-center lattices, only C center is acceptable. Hence, when the data are described in another center, it should be rewritten on C center, and specify `CELL_n` as `QUALIFIER`

4.2 Execution of `cryst`

Next, execute `cryst` by using data `*.xtl`. As an output, you will get `*.prim`. After this point, all the calculation refers `*.prim` as the crystal data, but not `*.xtl`. It may useful to remember that all the data in the calculation after `*.prim` are expressed in the atomic units. Only in output, some conventional units are sometimes used. In this case, explicit units will be given. Hence, if specific units are not given, the atomic units are assumed.

First place a file `si.xtl` in a working directory `/sidat`, then execute `cryst`. You are asked to answer

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

You should type as above, including a period at the last.

The output file `si.prim` should be checked.

Title and date appear, and the conventional unit cell, primitive unit cell, the reciprocal lattice vectors in the cartesian coordinates are followed, as

```
 TITLE GAAS
          date:     Fri Jan  5 17:56:20 2001
DIMENSION 3
 LATTICE PARAMETERS (A,B,C,CA,CB,CC) in a.u.
     10.6829045  10.6829045  10.6829045
      0.0000000   0.0000000   0.0000000
 Space group
      216   Td2     F-43m     ORIGIN_1
 IL     NG     NC
  2     24      1        ORI
 The conventional vectors
       10.6829045     0.0000000     0.0000000
        0.0000000    10.6829045     0.0000000
        0.0000000     0.0000000    10.6829045
 The primitive vectors
```

```
        0.0000000     5.3414522     5.3414522
        5.3414522     0.0000000     5.3414522
        5.3414522     5.3414522     0.0000000
 The primitive reciprocal vectors without 2Pi
       -0.0936075     0.0936075     0.0936075
        0.0936075    -0.0936075     0.0936075
        0.0936075     0.0936075    -0.0936075
```

IL, NG, and NC, which are parameters of TSPACE, are the type of lattice, the order of crystallographic point group, and the number of choices of the origin.

By using the space-group number, the crystal structure is constructed. It is important to check the output crystal structure. $N_{\rm spe}$ and all the chemical elements in the crystal are listed. Then, $N_{\rm ka}$ comes, and the Wyckoff positions of each sites are follows

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

After Wyckoff positions, $N_{\rm at}$ appears. After subtitle L.L. AND U.U., all $N_{\rm at}$ atoms are classified as $N_{\rm ka}$ cites, each line lists the range of those atoms belonging to an irreducible site. One line also contains the valence number, so that you should check it. Finally, relative coordinates of every atoms are listed.

Note

> On the outset, we should confess that quality of `cyrst` is less than other components of Osaka2k, in a sense of described below. `cyrst` attempts to determine Wyckoff positions by given coordinates and the space group information. Because 230 kinds of space groups are so complicated, this attempt in the present code does not guarantee to always success. For more details, refer to a supplement [**?**]. When it failed, `cryst` assigns the failed cite to a general point. Accordingly, user should take care of this assignment. If the assignment is wrong, you are required to fix manually. to a supplement [**?**].

Notations of Wyckoff positions accords to those of ITC. Its positional code accords to those of`TSPACE`. This is composed of three components. In each component is composed of an integer and a fraction number. The integer part means that `1`, `2`, and `3` means the relative coordinates x, y, and z, respectively, while `0` means just 0. When sign `-` are preceded to x, then negative value of x is meant. The fraction number are added to the integer part. For instance, `{1 0/1  2 1/4  -2 1/4}` means a coordinate $(x,\ y+\frac{1}{4},\ -y+\frac{1}{4})$.

4.3 Control parameters

So far, there is no case where users are required to modify parameters. For most cases, the default values of `TSPARAM` will suffice. But, if you want to calculate a large-size crystal, these default values may be insufficient. In this case, you are required to modify these by yourself.

`TSPARAM` are directory read and used in `TSPACE`, but these parameters are transformed explicitly or implicitly to other programs, such as `cryst`, `inip`, and `pwm`, and hence you should be care of setting these parameters.

```
PARAMETER (LMNATM=50,LMNKAT=20)
PARAMETER (MAXNPW=4854)
```

`LMNATM` specifies the upper limit for $N_{\rm at}$ in the unit cell, and `LMNKAT` specifies the upper limit for $N_{\rm ka}$. Therefore, it is recommended to take larger values than those actually used.

`MAXNPW` specifies the upper limit of the number of planewaves. But, the number of planewaves here is *different* one in `pwm`. Hence, you can set a value smaller than that used in `pwm`.

4.4 Graphics display of crystal structure

At this level, users must want to check correctness of crystal data by graphics, as well as numerics. Few people if not could imagine a crystal figure by numerics only, but most cannot do so without help of graphics.

There are many tools (commercial or public domain) to display crystal structures. Osaka2k offers a set of Mathematica notebooks to analyze results. Among them, `CrsytAnal.nb` analyzes the data `*.prim`. In order to display crystal structures, you are required only few input parameters, such as file name, in this Mathematica nootbook. Mathematica is very flexible, and I like this in data analysis. But, if you want to analyze further, for example, to calculate bond lengths or angles, you may need to knowledge for Mathematica.

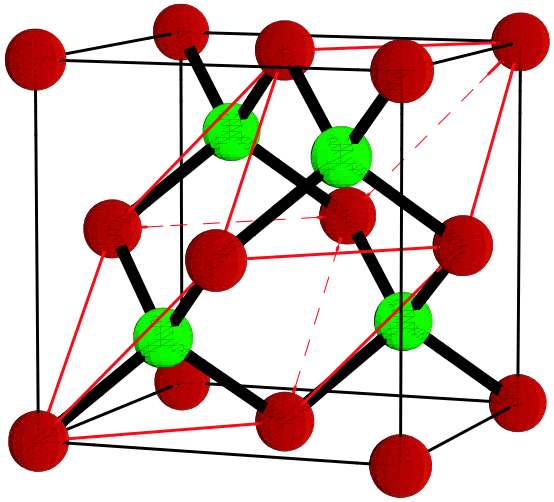

Figure 4.2: GaAs crystal. The red line indicates the primitive unit cell.

You will feel that commercial programs are easier to use than Mathematica nootbook. On the other hand, if you want to do further analysis, you may find after all that efforts as much as to understand Mathematica nootbook is required.

Although I do not here describe how to use Mathematica nootbook, those people who know the basics of Mathematica would not be difficult to understand that. Appendix A helps those people to use.

In Fig. 4.2, an example of crystal structure drawn by `CrystAnal.nb` is

shown.

Chapter 5

Ground states of electronic structures (I)

5.1 inip

Before proceeding to `pwm`, some preparations are needed. This is done by `inip`. [1]

One important purpose of `inip` is to expand the basis set of plane waves, given cutoff radius k_c in the reciprocal space. Count the number of plane waves $N_{\rm pw}$, and the cut the FFT box out. The relationships of the size of k_c to `NGDIM` and `NG3`, which determine the FFT box, are shown in Fig. 1.2. The length of one edge of FFT box is `2*NGDIM+1` in the k space, and the one in the real space `NADIM=2*NGDIM`.

Another purpose of `inip` is to determine k sampling point on the summation over the Brillouin zone, which is discribed in the next section.

A input file `inip.para` looks like as follows,

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

[1] In the previous version, the chief purpose of `inip` is estimation of the size of matrices appearing in `pwm`, in order to write them in a include file. In the current version, all the include files have been discarded, and accordingly such roles of `inip` is unnecessary. But, other things which are done by `inip` are still needed, and hence `inip` is left as being separated from `pwm`

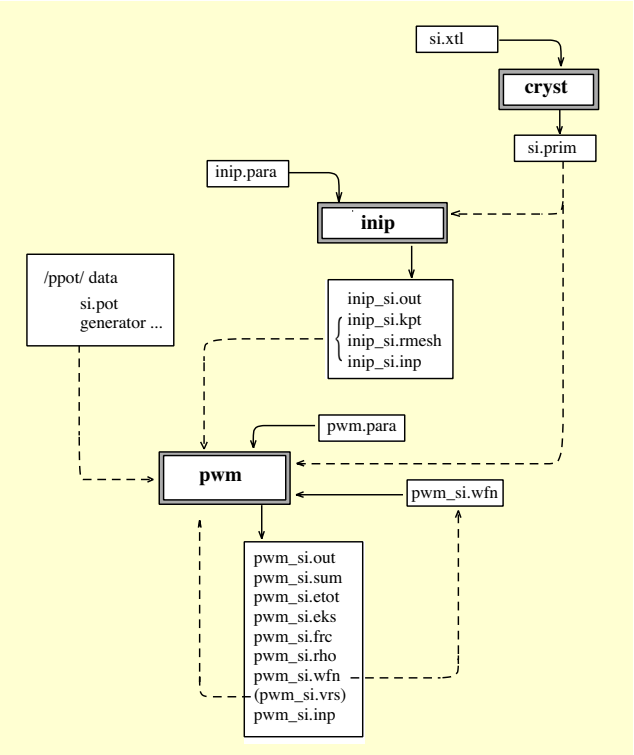

Figure 5.1: relationships among various files of `pwm`

```
1
number of k-sampling points
2
potential type (spin, NLCC, relativistic)
0         0         0
```

In the following, the meaning of these parameters are explained.

First the name of crystal comes with a period, which is followed by the cutoff radius for the plane wave expansion

- the cutoff radius of the plane waves (`AMAX`)
  the cutoff radius for the plane wave expansion k_c is given in the units of minimum value of the primitive reciprocal vectors $g_{\min}$.

- Way of giving k sampling
  1: automtic calculation, 0: manual input

- number of segments of k sampling (`NKDIV`)
  number of segments of k sampling when automatic calculation

- options for pseudopotential
  list of potential options. Three options are spin, core correction, and relativistic effect. 1 indicates ON, while 0 indicates OFF. These options are the same as those for pseudopotential-generation program `atom`, and thereby both must be the same.

Any information of potential type used `pwm` is inherited from the above parameters in `inip`. and accordingly no specification appears in `pwm.para`.

5.1.1 specail k-point sampling

There are several ways to giving k sampling points, as shown in the following.

(i) Automatic calculation
(isotropic and uniform sampling ≡ defualt)

When automatic calculation is chosen, k points are determined according to the method of Monkhorst-Park as described in Sec 1.6.[16] In default, the first zone is divided by M, indexing from $(-M+1)/2M$ to $(M-1)/2M$ in each of three principal direction in $2\pi/a$ units.

For example, when $M = 1$, only Γ is sampled. When$M = 2$, there are 8 points, *i.e.*, $(1/4, 1/4, 1/4)$ and ones changed with every sign. In a real crystal, further reduction of k points is achieved, because of crystal symmetry and time-reversal symmetry.

In Fig. 5.2, a feature of Monkhorst-Park is compared to that of beginner. A beginner would take first the center point, next the zone boundary points, and further proceed by taking the midpoint. On the other hand, Monkhorst-Park method avoids the zone center point and the boundary points as far as possible. This method gives better accuracy than that of beginners on the same computational task.

Even numbers for M is recommended, except when $M = 1$.

This value of M is given for input `NKDIV`. The determined sampling points and the total number is written out in an output file `inip_*.inp`, as

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

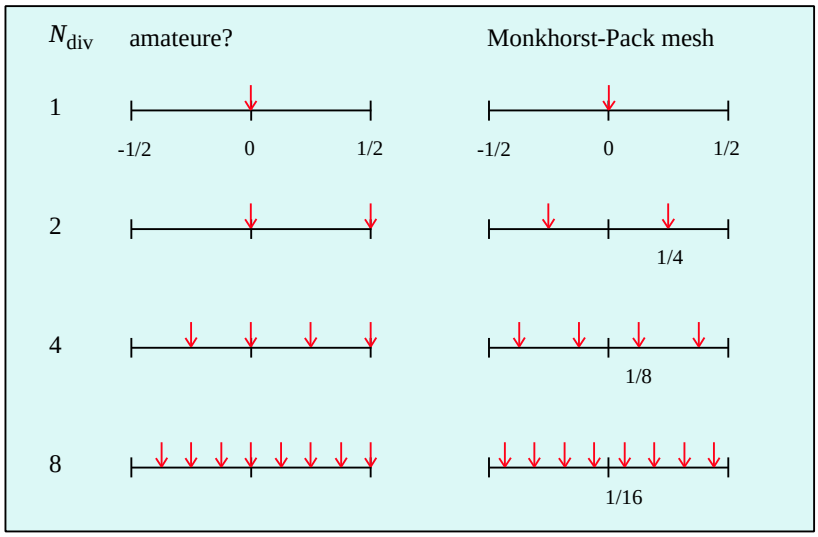

Figure 5.2: Comparison of k-point mesh between the method of Monkhorst-Park method and of amatua. In Monkhorst-Park method, the zone-center and zone-boundary points are avoided as far as possible.

In the above example, for an input $M = 2$, the final number of sampling points, n_{kpt}, only two. Remember the initial number of created k points is 8. Symmetry reduced greatly n_{kpt}. You can see that these points are $(-1/4, -1/4, -1/4)$ and $(1/4, -1/4, -1/4)$ in the primitive base, and the sampling wights are $1/4$ and $3/4$, respectively.

(ii) Automatic calculation (anisotropic sampling)

For crystals which are significantly anisotropic, such as super lattices, the above isotropic sampling is inefficient. In this case, anisotropic mesh is desirable. For doing so, optional parameters are available.

In `inip.para`, at the end of basic parameters, place one blank line, which is followed by

```
OPTION BEGIN
anisotrop_ksample=
  1   1   4   4
OPTION END
```

By declaring `anisotrop_ksample=`, the uniform division `NKDIV` previously given is made no effect. In the next line, the first three numbers mean the

segment numbers for the three primitive vectors $\mathbf{g}_1$, $\mathbf{g}_2$, and $\mathbf{g}_3$, respectively. The last one means the least common multiple (LCM) of first three integers. Since `inip` does not check correctness of this LCM, users should be careful for giving this number.

(iii) Manual input

Automatic calculation for giving k sampling point is convenient. On the other hand, there are cases in which a more flexible way to give sampling points is needed. In this case, user can specify sampling points manually. A user should determine sampling points by himself. The way of giving the points determined is as follows,

The portion from 6 to 9 lines in the above list for `inip.para` is replaced with

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

This example gives four points $(1/4, 1/4, 1/4)$ and $(1/4, 0, 0)$ and its cyclically changed points.

For the hexagonal system, other sampling method such as Chadi-Cohen algorithm [30] may be useful. In this case, the sampling points are given in this way.

When the number mesh points is small, a way of k sampling is critical. For example, when the total energies are compared among various polymorphic modifications, equivalent sampling method is important. Super lattices and polytypes of SiC are good examples [31].

In any method, it is desirable to estimate how used sampling is good. To do so, use of Eq. (1.33b) is made. In the section of `inip_*.out`,

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

you will find a list as

```
Quality of sampling by   2-points...
    list only nonzero shells up to  41 shells
    Sh No              shell sum
```

```
 8     (  -1.00000,   0.00000)
15     (   1.00000,   0.00000)
23     (  -1.00000,   0.00000)
26     (  -1.00000,   0.00000)
28     (   1.00000,   0.00000)
32     (  -1.00000,   0.00000)
36     (   1.00000,   0.00000)
38     (   1.00000,   0.00000)
```

This is a list of only those shell sums in Eq. (**??**) which gives no zero. In the example shows that the shell sum vanishes up to 7th shell.

5.1.2 Cutoff radius of planewaves

Another important parameter is the cutoff radius (`AMAX`) for plane wave expansion. The dimensionless input parameter `AMAX` is related to the cutoff radius $k_{\rm c}$ in the k space by

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

where $g_{\rm min}$ is the minimum length of the three primitive reciprocal vectors. The cutoff energy $E_{\rm cut}$ is given by Eq. (1.25) in the atomic units.

For each sampling point $\mathbf{k}_i$, all the plane waves $\mathbf{G}$ which satisfy Eq. (1.24) are taken as the basis set. Accordingly, the number of plane waves $N_{\rm pw}$ differs slightly from point to point.

In an output file `inip_*.out`, you will find an example as

```
=====> Final report for k-point sampling <=====
For cutoff energy in Ry =   4.959185
NHDIM =     59
NGDIM =      4
nkpts=   2
Name: LD   -1 -1 -1/  4    NSTR=  8   WTK= 0.25000    NPW=  55   INV= 6
Name: XY    1 -1 -1/  4    NSTR= 24   WTK= 0.75000    NPW=  50   INV=18
          Sum over WTK  1.000000
```

From this, you see that point $(-1/4, -1/4, -1/4)$ and $(1/4, -1/4, -1/4)$ have different $N_{\rm pw}$, 55 and 50, respectively. The basis set of these points are stored in an output file `inip_*.kpt`. `inip` determines the dimension of the array of basis set (`NHDIM`) as the maximum of these $N_{\rm pw}$'s, *i.e.*, 59.

Along with `NHDIM`, parameters used for `pwm` are summarized in the last section of `inip_*.out`, as

```
Passed parameters
          NEPC          8          NEDIM         4
```

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

As another role of `inip` creates a list of real-space points, which are connected by the crystal symmetry. This data is stored in a file `inip_*.rmesh`. When there is no symmetry, *i.e.*, the space group is P1, this file is not created.

5.2 pwm

The core program of Osaka2k is `pwm`. This program calculates the electronic ground states for a given crystal. This is achieved by self-consistent energy minimization for the energy functional as described in Sec. 1.7.

After creating an executable code `pwm` as described in Sec. 2.2, make a copy or link `pwm` into your working directory.

5.2.1 SCF calculation

Input parameters for `pwm` is written in `pwm.para`. There are a couple of parameters, which control calculations. As described earlier, we ceased to pack all the input parameters in the standard format of `pwm.para`. Instead, rarely used parameters are written by optional input, when need occurs.

A typical form of `pwm.para` looks like as follows,

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

```
2
```

In the following, meaning of each parameters are explained.

`xtl.name`	name of crystal
`JobNo`	Job type 0 : SCF calculation (default) 1 : optimization of atom positions 2 : optimization of unit cell 4 : MDS
`atomMove0ctrl`	atom relaxation 0 : no (default) 1 : do
maxIter0wfn	the maximum number of iterations for energy minimization process with respect wave functions
`npath0wfn`	iteration number of conjugate-gradient paths for a band
`wftol`	tolerance of wavefunction error (Ry^2)
`ftol`	tolerance of HF forces ($\mathrm{Ry}/{a_B}^2$)
`etol`	tolerance of energy (Ry)
`iread_wfn`	selection of initial wave functions 0: random data (default)
`imoni0ctrl`	monitoring 0 : OFF (default)
`irhout`	output of charge density 1: ON (default)
`iwfout`	output of wave functions 1: ON (default)
`atomMove_resume`	resume of atom relaxation 0 : OFF (default)
`maxIter0atom`	the maximum iterations for atom relaxation
`cellMove_resume`	resume of cell relaxation 0 : OFF (default)
`maxIter0cell`	the maximum iterations for cell relaxation

The most important parameters are those of convergent criteria. As described in Sec. 1.7, the electronic ground state can be found by iterative minimization of the total energy. As the smallest iteration loop, the minimization process by the conjugate-gradient method is applied to one band at a $\mathbf{k}_i$ point. For a band, the conjugate-gradient path is repeated by `npath0wfn`. Then, this process is repeated over the all bands of `nband` at a $\mathbf{k}_i$ point. After completing this at $\mathbf{k}_i$, proceed to the next $\mathbf{k}_j$ point. After repeating over all the k sampling points, one iteration in the most outer loop is completed. This outer loop will be repeated by `maxIter0wfn`, if convergence criteria are not satisfied.

Among these parameters, the most frequently changed may be `maxIter0wfn`. This value may be chosen to be about 5. SCF calculation will stop, when the iteration of the outer loop exceeds `maxIter0wfn`, or when the error of

wave functions becomes less that `wftol`, or when energy variation from the previous step becomes less that `etol`

If you intend to achieve SCF calculation of electronic ground state only, `JobNo` is set to be 0. In this case, all the remaining parameters after `atomMove_resume` are ignored.

For a given crystal, at the first time of calculation, set `iread_wfn=0`. In this case, the initial wave functions are created by random data.

As output control, there are three options to output files, monitering (`imoni0ctrl`), charge density (`irhout`), wave functions (`iwout`). Usually, set `imoni0ctrl=0`. Only for debugging, this will be changed. When treating a large crystal, and if you do not need further calculation than SCF calculation for the fixed crystal, then you can save memory not to output wave function file by setting `iwout = 0`. It is noted that, to save memory, wavefunction data is written in binary file. Hence, this file is usable only on the working machine.

On completing the above preparation, you can run `pwm`. When short calculation, directory type

```
% pwm
```

After launching `pwm`, the process of calculation is flowing out on the display. When long calculation, it should be put on as a background job, by

```
% pwm>stdout &
```

Because `pwm` prints out some output directly on the display, the above redirection for output is needed, otherwise causes output error. When the job is put on a queue system, keep instructions of the system.

It is important to estimate the calculation time for a given crystal. As a guide, we include benchmark tests in a directory `bncmrk`. This was done for a crystal containing 8 Si atoms. Various platforms were examined. You can acquaint some idea about how long calculation is taken.

5.2.2 Interpretation

On completing calculation, `pwm` outputs various files as shown in Fig. 5.1.

pwm_*.out: log record

pwm_*.eks: KS levels

pwm_*.frc: Hellman-Feynman forces

pwm_*.etot: total energy

pwm_*.sum: summary

pwm_*.rho: charge density

pwm_*.wfn: wavefunction (binary format)

Among them, the most important one is `pwm_*.etot`. The file begins with a description of calculation type and conditions, as

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

JOB Type:
     Self-consistent Electronic Calculation
Potential Type:
     spin = 0      nlcc = 0      relativ = 0
    KLEINMANN-BYLANDER TYPE FULLY SEPARABLE FORM
    Ceperly-Alder type electron-correlation functional
    maximum l =     1
...
```

A description of crystal follows. User should check these descriptions.

After the title of `SCF calculation`, the file describes the essence of iterative minimization of the total energy,

```
==================== SCF calculation ========================

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

 .....

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

- iter: step number of iteration

- $E_{\rm el}$: the energy of electrons $E_{\rm el}$
- ΔE: change in $E_{\rm el}$ from the previous step
- ξ: residual error in wavefunction, Eq. (1.40)
- the number of paths actually carried out/ the number of back path: usually the former number is equal to the set value `npath`. But, at sometime, the conjugate-gradient paths are stopped before reaching the `npath` step, because of satisfaction of other criteria of convergence. At sometime, the minimization process fails by some reasons. The electron energy may be increased. If this happens, then the conjugate-gradient process for this band is stopped, and the number of such an event is counted as the number of back path.
- θ: the maximum value of mixing angles of all the bands, θ, in Eq. (1.45)

As shown in the example, as the process proceeds, $E_{\rm el}$ is decreased, accordingly the absolute of ΔE, ξ, and θ are also decreased. The process is stopped at step 5, because$|\Delta E| <$ etol. Then, the total energy $E_{\rm tot}$ is printed at the last by adding the electrostatic energy of ions $E_{\rm ewald}$ to the converged electron energy $E_{\rm el}$.

Next, decomposition of the total energy $E_{\rm tot}$ is listed as

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

This decomposition is given by Eq. (1.22). From this analysis, which contribution dominates the total energy may be investigated. However, this way of decomposition is somewhat arbitrary or is based on mathematical convenience. Hence, users should remain that to place too much physical meaning on them is dangerous.

For example, although the ion-ion electrostatic interaction energy, *i.e.*, Ewald term Eq. (1.23) seems to be most definite, it cannot be so clearly separated from other terms. In Ref. [23], it is shown that the Ewald term is decreased, when a Si crystal is compressed. Because all the ions in Si crystal have +4 charge, it will be expected that the electrostatic energy should be increased as the inter-atomic distance is decreased. This seemingly unexpected result is a consequence of the fact that even Ewald term is not

independent of other interaction terms such as ion-electron interaction in a form of Eq. 1.22. [27]

At the last, stress components are listed as

```
total stress (Ry/Bohr^3)
      S(1,1)=    -2.24759E-05      S(2,3)=     3.17724E-10
      S(2,2)=    -2.24759E-05      S(3,1)=     2.52820E-10
      S(3,3)=    -2.24756E-05      S(1,2)=    -1.10795E-10
```

By evaluating stresses for a given strain, we can calculate elastic constants. The total stress also can be decomposed as for the total energy. This is done in `pwm_*.strs`. However, some cautions are needed in the interpretation as for the decomposition of the total energy.

For example, the first term of the total energy in Eq. (1.22) represents the kinetic energy. The corresponding component of the total stress is given by

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

in Eq. (2) of Ref. [23]. This correspondence is clear. But, if you interpret the above expression 5.2 as indicating the change exactly in the kinetic energy component in the total energy given by Eq. (1.22), it would be wrong. I means that the two is numerically different. Hellmann-Faynman theorem guarantees that the *total* force or stress are given by the change in the *total* energy if the true ground state is found, but does not guarantees that each component of force or stress are given by the change in the corresponding component of the total energy. This equality holds only for the whole, but not for individual components.

Hellmann-Feynman forces for each atom are listed in `pwm_*.frc`, as

```
HELLMANN-FEYNMAN FORCE  (RY/BOHR)
            ATOM                   X               Y               Z
ITER=    1
              1   1    -0.230775E-08  -0.154249E-07  -0.573629E-07
              1   2     0.637597E-07   0.428339E-07   0.299777E-07
```

These three components are expressed in the Cartesian coordinates, which are those of used in constructing the crystal.

Accuracy of energy

In the above example, the electron energy $E_{\rm el}$ is well converged within $\Delta E_{\rm el} < 10^{-8}$. User may be surprised to see that the value is in a good agreement with

each other among various platforms. This is a good news. Unfortunately, this does not mean that this accuracy gives smallness of the error in $E_{\rm el}$ from the true value. This merely indicates the numerical stability, even if the error from the true value is large.

The error analysis of $E_{\rm el}$ is a complicated problem. This is an author's suffering not to be able to provide the definite criteria as to which extent the magnitude of error changes for given conditions.

Let us to evaluate errors in `pwm` from a viewpoint of general error analysis in numeric calculations. When referring errors in $E_{\rm el}$, first it must be clarified what kinds of errors are involved. On input, errors (i) in input data can occur. During calculation, errors of (ii) rounding off and other inherent ones in digital computations are unavoidable. Errors are also introduced (iii) by approximations used in the calculation. All the errors are more or less amplified to yield the final error. This process is called *error propagation*, and is an important concept in numerical analysis, along with *numerical stability*.[29]

In the present analysis, forget the first kind of error, *i.e.*, input error, such as errors in a crystal structure, potentials, etc.

Let us estimate errors of kind (iii), especially, descretization errors in evaluating the total energy expressed by Eq. (1.22). For simplicity, only a term of ion-electron interaction is considered. It is given by an integration over the real space, $\int \rho(\mathbf{r})V(\mathbf{r})d^3\mathbf{r}$. The integration is, of course, evaluated by summing over the finite segmentations. When the integration $\int_{-h}^{h} f(x)dx$ on the small finite range $\{-h, h\}$ is replaced merely by $2hf(0)$, the error is estimated by

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

In the integration $\int \rho(\mathbf{r})V(\mathbf{r})d^3\mathbf{r}$, the integrand is given by a product of the potential and the charge density. Here, we simplify a problem by taking only a potential; ρ has merely an effect to smear the effective potential. For Si crystal, the integration may be dominated by a contribution from the bond center, the value of potential and its second derivatives at the region are assumed in evaluation of Eq. (5.3), then they yield the coefficient of h^2 to be ~ 0.05. the width of mesh in the real space h is given by a_0/NADIM, where `NADIM` is the number of descritization for the lattice length $a_0 \sim 7$. When tt NADIM=16, the relative error becomes 1×10^{-2}. When `NADIM=40`, the relative error is barely improved by one order of magnitude.

By using a Si crystal, actual calculation is carried out by changing $E_{\rm cut}$ to see the effect on the accuracy of $E_{\rm el}$. In Table 5.1, the results are listed.

As shown in the Table, when `NADIM=16`, the value $E_{\rm el}$ has two digits of accuracy under the decimal point. When `NADIM=40`, it has four digits of accuracy (Ry/cell units). By considering the above assumption, the agreement between expectation and the result is good. According to Eq. (5.3), three times fine mesh is required to obtain the accuracy of $E_{\rm el}$ by one order of magnitude, since the relative error is

Table 5.1: Relationship between $N_{\rm pw}$ and $E_{\rm el}$ for Si.

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

proportional to h^2. But, the calculational task increases as the cube of $N_{\rm pw}$. [2]

Anyway, according to Eq. (5.3), as potential variation is more rapid, we have larger descritization errors. This means that the convergence with respect to $N_{\rm pw}$ gets worse, when steep variation of potential.

Now we have seen how large the discretizing error is. Then, a question arises as to whether further digits below the accuracy limit is in all meaningless or not. As the absolute value, it is true. However, for a relative comparison, even further digits make sense. Actually, in many cases, we are interested in relative comparison of the total energy under various situations. If descretizing errors occur in a similar way in such situations, then further digits still have meaning. If calculations of various crystal forms are carried out under the same cutoff energy in plane wave expansion, this is indeed the case. In this case, what determines the *relative* accuracy of the final result is a problem of error propagation (ii), *i.e.*, numeric stability.

Referring to [29], given an input x, we calculate the final result y by $y = \varphi(x)$. If the relative error in the input x due to, for example, machine accuracy, is expressed by ε_x, then the relative error in the output ε_y becomes

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

In Eq. a coefficient (5.4), $(x/\varphi(x))\partial\varphi(x)/\partial x$ represents the amplification of the error, and called *condition number*. Let us estimate the condition number in this example. For a fixed cutoff energy of planewaves, by slightly changing the atom position x, let us examine how the total energy $y = E_{\rm el}$ is influenced. In a crystal Si_8, atom at the equilibrium position $(0, 0, 0)$ is moved in the x direction.

$E_{\rm cut} = 13.94$ Ry and $N_{\rm pw} = 958$ are used.

[2] In evaluation of integral in the total energy, higher-order evaluation like Simpson formulus may be an efficient method to avoid the cube dependence on $N_{\rm pw}$.

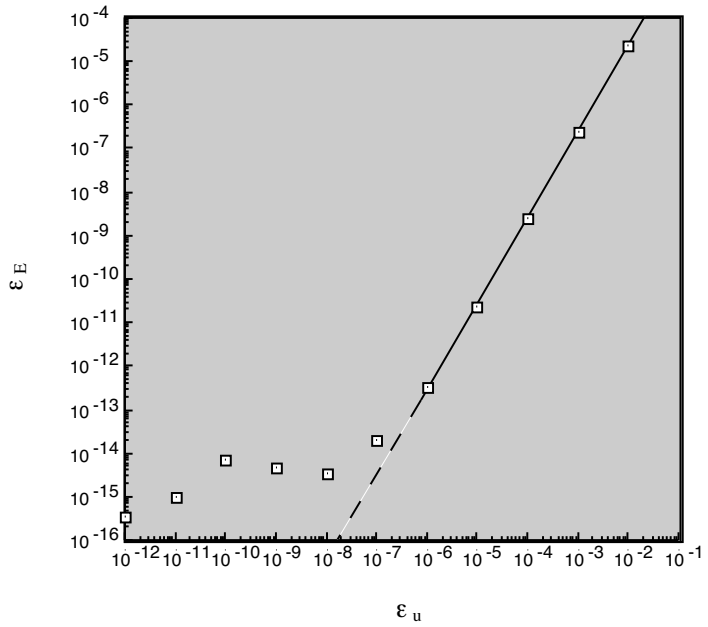

Figure 5.3: The relative error in the total energy, ε_{E}, of Si_8 for the input error ε_u due to error of atom positions.

The result is shown in Fig. 5.3. From this figure, you can see that the condition number is small enough over a wide range of error in u. This means numerical stability. By the relation of ε_{E} with respect to ε_u larger than 10^{-7}, we can evaluate that $\varepsilon_{\mathrm{E}} \sim \varepsilon_u^2$. This reflects that the variation of potential from the equilibrium potision is approximated by harmonic form.

In this way, we can see that because of good numeric stability of `pwm`, we can compare the total energy exceeding two digits, if the same computational conditions are used. The problem of energy accuracy is further discussed later (in Sec. 6.1.3).

5.2.3 Recalculation

If you find that one SCF calculation is not enough to achieve good convergence, then you can continue the calculation. In this case, the output data of wavefunction is used as the input of the next calculation. User are required

to rename the output file `pwm_*.wfn` as `inip_*.wfn`, and set `iread` to be 1 in `pwm.para`. Then, the next calculation will start with this wavefunction as the initial state.

Estimation of various energies

How can the total energy obtained in this way be interpreted? We have evaluated the total energy by the pseudopotential method, *i.e.*, Eq. (1.22). Therefore, only valence contribution is involved. Fortunately, most of chemical properties of solids are determined by valence electron solely. Many interested energies can be evaluated through evaluation of the total energy by pseudopotential method.

For example, the cohesive energy $E_{\rm coh}(A)$ of element A can be estimated by

$$E_{\rm coh}(A) = E(A^{\rm (gas)}) - E(A^{\rm (sol)}). \tag{5.5}$$

The formation energy $E_{\rm form}(A_{\rm m}B_{\rm n})$ of compound $A_{\rm m}B_{\rm n}$ can be estimated by

$$E_{\rm form}(A_{\rm m}B_{\rm n}) = mE(A^{\rm (sol)}) + nE(B^{\rm (sol)}) - E(A_{\rm m}B_{\rm n}). \tag{5.6}$$

When comparing the energy between solid and isolated atom, it is known that there is a problem regarding continuity of the exchange-correlation energy. This is not a problem of pseudopotential approximation, but essential problem of DFT. There are many discussions in the literature, but here only one of early studies by pseudopotential is cited.[28].

5.2.4 Display of charge density

After you can complete `pwm` without trouble, the first quantity which you can see is the total energy. But, the energy itself is relative quantity, in particular, so in pseudopotential mathod. Even using the same pseudopotential method, and even using the same pseudopotentials, you can see different values between different implementations (different codes). It is almost impossible to justify if the obtained value of total energy is reasonable, if you see the total energy only. In most cases, to see the charge distribution is a good test for justifying the correctness of calculation.

The charge density in the real space is written out in file `pwm_*.rho`. The format of `pwm_*.rho` is that at the top the potential types (`spin, NLCC, relativistic` in input file `inip.para`) and the numbers of segmentations (`{NAX, NAY, NAZ}`) in the real space are described as

```
       0       0       0
      16      16      16
spin= -1
```

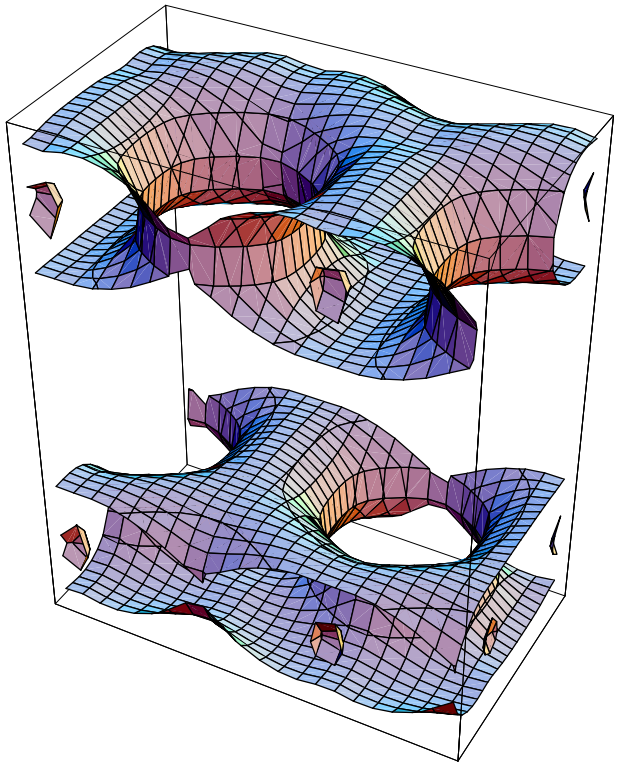

Figure 5.4: A surface of equi-density of charge for graphite

After spin identification, the charge density $\rho(\mathbf{r}) = \rho(i, j, k)$ is sequentially written in order that i is taken as the most inner loop, while k is as the most outer loop.

For visualization of charge density, a Mathematica notebook`ChgDnst.nb` can be used. As input, a crystal data `*.prim` and the charge density data `pwm_*.rho` are used. As an example, three-dimensional plot of charge density of graphite is shown in Fig. 5.4.

From the figure, you can see that honeycomb bonding is formed in a layer. In this way, three-dimensional display is useful for quick grasping how the charge is distributed. Although appealing appearance, three-dimensional display has a disadvantage in less quantitative than two-dimensional contour map of charge distribution. An example is shown in Fig. 5.5. There are two C atoms in a plane $z = 1/4$. The thick line indicates the boundary of

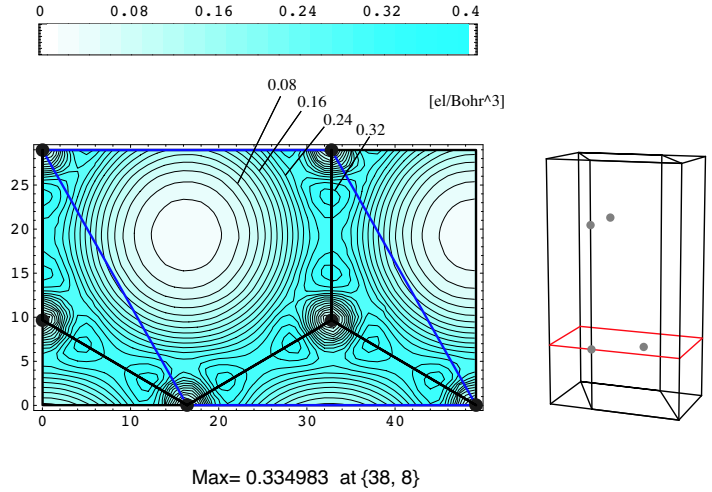

Figure 5.5: A contour map of charge density of graphite, on a cut plane $z = 1/4$

the primitive unit cell, and gray points indicate atoms. The contour map is drawn so that the more dark part is, the higher density it means. You can see that a strong covalent bond is formed between two adjacent atoms. Two-dimensional plotting has a disadvantage to show only one cut plane in a plot. Hence, you should be careful not missing the whole picture.

More details of `ChgDnst.nb` is given by Appendix A.

Correctness of obtained charge density is judged after all by the discipline and experience as physicist, Then, you may say that we rely on calculations just because we do not know how the charge is distributed. If we know that, we would not need calculations. Yes, you are right. Even for experienced theoreticians, it is difficult to judge correctness of two different distributions, if difference is only quantitative. We can only say that it is just beginning of physics. In addition, even if your result is perfect in terms of calculational technique, you should remind that the current status of the first-principles calculation is still far from perfect. We are facing to electrons of $\sim 10^{23}$. I like to close this section with a wise man's saying, although it is practically no help for the above question,

> "This is not to say that it solves the many-body problem; no approach does that – if one miraculously did, physics would be much less interesting and challenging." (Callaway), Ref. [32], p. 106

5.2.5 Cases of wrong convergence

In most cases, SCF calculation is completed with sufficient convergence. But, you may encounter that it is not so. In case of odd-number of electrons, this may happen. The top of valence band is occupied only partially. In the line minimization process of the conjugate-gradient method, the prediction based on parabolic extrapolation is not guaranteed to succeed. In addition, crossing of nearly degenerate levels can happen, which causes instability to the minimization process.

This can, in particular, be serious for metals. In metals, the valence top occurs at somewhere of the Brillouin zone. Gapless feature of metals causes alternation of occupancy at different k points, which makes the minimization process unstable.

There is no universal prescription for the above problem. We can only do case by case, some of which are found in Technical Report Series. Here, basics what we should do are described below, based on my experience.

On the outset, it should be reminded that it does not necessarily become unstable simply because of metals. Don't attribute instability always to the odd-number of electrons, whenever it happens. In my experience, even for metals, we can get good convergence, unless degeneracy occurs at the top of bands or between different k points.

Then, when you encounter difficulty in convergence, first check whether it is caused by degeneracy at the top valence levels. In input file `pwm.para`, add an option,

```
allow_posE ON
```

Watch how the top levels are varied as the minimization proceeds. Check if oscilation of specific band or alternation among several levels occur.

At this point, it should be care of concept of degeneracy

1. Rigorously speaking, degeneracy is predicted by symmetry reason. However, in actual calculations, the criterion for degeneracy is made only by numerical estimation. Hence, numerical accuracy becomes problem. Usually, those levels which fall within some energy range can be regarded as being degeneracy.

2. Degeneracy occurs not only for those bands at a k point, but also for bands at different k. Convention of the band theory is that degeneracy between different k points are not regarded as degeneracy. But, in

the actual calculation, this kind is still degeneracy, which yield same difficulty as the first kind.

In `pwm`, bands are updated by band by band. During this process, those occupation numbers are fixed. If one or some of these levels exceed the Fermi level, the line minimization based on parabolic extrapolation would be failed. When this is the case, a method of broadening of occupation number is generally useful. In `pwm`, Fermi broadening method is available.

Other cases of wrong convergence

You may encounter convergence problem other than the previous type. For example, the line minimization could be failed even when occupation is not problem. The line minimization is based on the parabolic extrapolation. See Eq. (1.46), that the function is given by a quadric with respect to θ. Usually, this treatment works well. But, because of nonlinearity of the exchange-correlation functional with respect to the variation in density, this approximation could give bad a prediction for the minimum position.

5.2.6 Related options

There are many options for controlling SCF calculations. They may be changed from version to version.

- diagonalize OFF
  By default, diagonalization of Hamiltonian is performed within the subspace of the valence electrons at the last of each step of the energy minimization. By declaring in this way, the diagonalization process is omitted.

- allow_posE ON
  By default, the energy minimization process is stopped if the change $\Delta E_{\rm el}$ is increased. By declaring in this way, the process is continued even if positive $\Delta E_{\rm el}$ is met.

- ran_wfn=
  1
  The type of initial guess for wavefunctions. By default (1), it is created by random numbers. Set 0, if wavefunctions constructed by specific plane waves are used.

- vary_occ ON
  Make the occupation number variable. By default, it is fixed.

- energ_deg_Ef=
  1.0E-4
  The energy criterion if energy levels are degenerate.

- nel_add=
  1
  Electrons are added (subtracted if negative number) by a given number (integer).

- rnel_add=
  0.9
  Electrons are added (subtracted if negative number) by a given number (real number).

- nband_extra=
  1
  Bands are added by a given number. By default, the number of band is half of the number of all the valence electrons.

- mix_occ=
  1.0D-1
  When the occupation number is varied, weight of mixing of new and old wavefunctions. 1 means use of completely new wavefunction (default).

- fermi_broadening ON
  The occupation number is determined by Fermi-Dirac distribution function.

- T_elec_start0fermi=
  0.1
  Initial electron temperature when Fermi-broadening is applied.

- T_elec_end0fermi=
  0.1
  Distinated electron temperature when Fermi-broadening is applied.

You can see that many of options in this section are related to the occupation. Examples of real calculations are found in Technical Reports, as

- No. 14 "Control of the number of electron – with application to energy gap –"

- No. 17 "Impurity levels in semiconductors"
- No. 37 "Manage of electron occupation in case of odd-number electrons"

Chapter 6

Ground-States of Solids (II)

Once `pwm` completes a SCF calculation, you have several choices on which we work.

6.1 Atom Optimization

The total energy $E_{\rm tot}$ of a crystal is a function of atom positions $\mathbf{R}_i$. If the primitive unit cell contains N atoms, $E_{\rm tot}(\{\mathbf{R}_i\})$ is a function of $3N$ variables. The equilibrium positions of atoms are such a configuration that the $E_{\rm tot}(\{\mathbf{R}_i\})$ takes minimum with respect to atom positions. Usually, the energy function has many local minima, so that to search the global minimum is difficult. Indeed, no algorithm can locate the global minimum in general situations.

In this chapter, we restrict ourselves to search a local minimum on a linear theory. Hence, the obtained minimum is highly dependent of the initial positions. While many algorithms are available for this purpose, we employ the conjugate-gradient method based on linear approximation. By saying *linear*, it is meant that the line minimization along the conjugate-gradient direction is carried out by parabolic approximation Becase in `pwm`, Hellmann-Feynman forces, i.e., energy gradient with respect to atom positions can be easily evaluated, this method is especially effective.

By using the conjugate-gradient method, all the atoms in the primitive unit cell can be moved to obtain the optimal structure. When we keep the original crystal symmetry, the atom movements are restricted in order not to violate the crystal symmetry. In other words, the atom can be displaced in a range determined by Wyckoff positions. Therefore, for example, for Si of the original space group O_h^7, all the atoms cannot be displaced by this

method. Only when the crystal symmetry is artificially lowered, *e.g.*, to P1, the atoms can be displaced.

6.1.1 Preparation of calculation

Similar to the procedure described in Sec. 5.2.3, the output wavefunctions (`pwm_*.wfn`) of SCF calculation should be prepared. Then, input file `pwm.para` is edited as

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

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

ftol
8.0D-5
.....

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

First, select 1 as `Job Type`. Set `maxIter0atom` an appropriate value, which is the upper limit for the iterations of conjugate-gradient process for atoms.

In this case, the maximum iterations for wavefunction, `maxIter0wfn`, can be reduced from that of SCF calculation. To use the previous wavefunction data, rename it to `inip_*.wfn`, and set `iread` 1. Now, we are ready to run `pwm`.

Calculation will be ended when iterations of atom movement exceed `maxIter0atom`, or when the residual force becomes less than `ftol` (Ry/a_{B} unit).

When a calculation does not end with a sufficient relaxation, you can continue that calculation, by setting `Resume atom relaxation = 1`. In this case, the atom positions are succeeded from those of which the previous calculation achieved. This is done by reading file `pwm_*.var`. So, you don't need to rewrite file *.prim. For doing so, the data file `pwm_*.var` should be renamed as `inip_*.var`. As to wavefunction, of course, rename the previous output file `pwm_*.wfn` to `inip_*.wfn`.

6.1.2 Interpretation

As an example, we perform an atom optimization for a crystal Si_8 whose atom positions are slightly deviated from the equilibrium positions.

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

As listed above, the relative coordinates are deviated at three digits under the decimal point.

Let us apply the conjugate-gradient process to this by five times.

```
ATOM
ITER          Etot            delE           rsforc        frc_grd           umin
====    ==============    =============    ============    ============    ============
   0     -63.22541053      0.000E+00        2.446E-02      -1.957E-01       0.000E+00
   1     -63.27078260     -4.537E-02        5.633E-03      -4.395E-02       4.516E-01
   2     -63.27326505     -2.482E-03        2.116E-03      -1.578E-02       1.127E-01
   3     -63.27381923     -5.542E-04        1.847E-03      -1.079E-02       6.977E-02
   4     -63.27456888     -7.496E-04        1.613E-03      -8.261E-03       1.391E-01
   5     -63.27478859     -2.197E-04        4.204E-04      -3.124E-03       5.204E-02
```

This lists the step number of the conjugate-gradient process of atom positions, the total energy tt Etot, the change in the total energy `delE`, the residual error in the wavefunction `resid`, residual Hellmann-Feynman force (its average `rsf` and its projection onto the searching direction `frgr`, and displacement per atom `dx/at`. As seen, the total energy, residual forces, and atom displacements are monotonously decreased, as expected.

As a result, the final values of the relative coordinates become

```
Positions relative to the conventional unit cell
   1     0.0036562     0.0002995    -0.0061577        1    si     1
   2     0.0038705     0.5001516     0.4937230        1    si     1
   3     0.5035348     0.0000587     0.4935374        1    si     1
   4     0.5039569     0.4997671    -0.0063730        1    si     1
   5     0.2537480     0.2497998     0.2440929        1    si     1
   6     0.7537595     0.7500895     0.2440021        1    si     1
   7     0.7536154     0.2498403     0.7436120        1    si     1
   8     0.2538493     0.7499966     0.7435759        1    si     1
```

These are in agreement with the equilibrium value by three digits under the decimal point.

Here, I like to pull user's attention regarding Hellmann-Feynman forces. It should be that the sum of all the forces in the primitive unit cell is zero. In other words, the sum of inner forces must be zero. One consequence of this is that the displacement of atom positions is allowed up to rigid translations. Of course, in our case, because of numeric computation, there are always numeric errors, so that the sum of forces never vanish. Hence, how the sum of forces deviate from zero is a test of numerical accuracy. You can see this sum in file
pwm_*.frc. After listing Hellmann-Feynman forces of all the atoms, the sum is shown as

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

This magnitude is good enough for usual purpose.

6.1.3 Error in forces

Mathematically, the residual Hellmann-Feynman forces can be reduced by arbitrary small. In practice, once the residual forces are reduced to a some level, further iterations do not improve accuracy any more, because of numerical errors.

Let check the error in forces more closely. As discussed in 5.2.2, when we refer to error estimation in daily calculations, what we can usually do is to estimate relative errors, but not the absolute error. This means that we ignore the error in the total energy obtained by using a truncated number of the basis set from that value by using complete set. Within a fixed number of the basis set, we estimate uncertainty of the total energy ΔE. We do this usually by seeing the energy change ΔE in the last step of the iterative processes. Let assume a harmonic approximation between the total energy $E_{\rm tot}$ and the deviation of atom from its equilibrium position Δx,

$$\Delta E = \frac{1}{2} k (\Delta x)^2, \tag{6.1}$$

where k is a force constant between a pair of atoms under consideration. Uncertainty in $E_{\rm tot}$ leads to the error in position Δx by $\sqrt{2\Delta E/k}$, *i.e.*, Δx is not linear against to ΔE but its square root. Hence, uncertainty in position is larger than that of the energy. By using a relation $f = kx$, *i.e.*,

$$\Delta f = \sqrt{2k\Delta E}, \tag{6.2}$$

we can conclude that the error in force is more severe than that of ΔE.

For example, let us consider Si crystal. In the direction of bond axis, the force constant is $k_r \simeq 0.3\,[\mathrm{Ry/Bohr}^2]$. The energy change at the last step of SCF calculation with respect to wavefunction is order of 1.0^{-7} Ry, and by Eq. (6.2) we have an error of force, $\Delta f = 2.5 \times 10^{-4}$. From this, uncertainty in position is estimated as $\Delta u \sim 1 \times 10^{-3}$, which corresponds to $\sim 1 \times 10^{-4}$ in the relative unit. This says that relative coordinates are accurate up to four digits under the decimal point. You can confirm in the above output file for atom relaxation that the accuracy of relative coordinates are in this order.

6.1.4 Convergence problem

As usual, behind advantages of the conjugate-gradient method, some risks are hindered. Here, some cases in which the conjugate-gradient process fails are discussed.

The most prominent feature of the conjugate-gradient method, compared with the steepest-descent method, resides in a way of determining the searching direction in each step of minimization. Once the searching direction is determined, then the minimizing process in the one-dimensional space is the same between these two methods. Usually, parabolic approximation is used, that is, the energy function is approximated in this direction by $E(x) = ax^2 + bx + c$. We already have the function value (energy) E_0 and its derivative $E'(0) = -f$ at a set of atom positions $x = x_0$. Without loss of generality, we can take x_0 as the origin. Only another piece of information is enough to determine completely the parabolic form $E(x)$. Naturally, we obtain another one by evaluating $E(x)$ at another point $x = x_1$, *i.e.*, E_1.

Thus, the position of energy minimum $x_{\min}$ is determined by

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

where a and b are

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

respectively.

Hence, at the cost of minimum computational task to locate the minimum point in a searching direction, it applicability is limited by parabolic

approximation. [1]. For most cases, the parabolic approximation is not so bad, but sometimes fails.

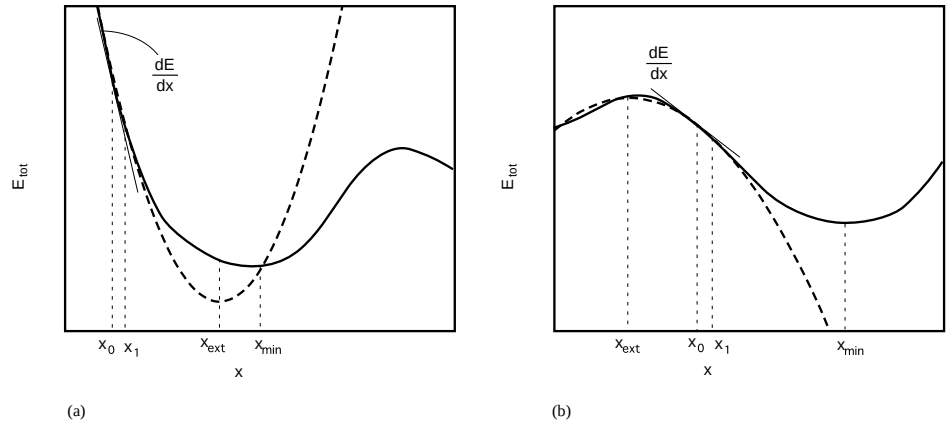

Figure 6.1: An example of failure in the parabolic extrapolation. The solid line indicates the true adiabatic potential, while the dashed line indicates the parabolic approximation

First example of the failure is shown in Fig. 6.1 (a). As shown in the figure, the extrapolated point $x_{\rm ext}$ by using evaluation at x_0 and $x = x_1$ is largely deviated from the true minimum point $x_{\rm min}$. Even in this case, if we take the next step from $x_{\rm ext}$ as the next starting point, we could reach the true minimum point. But, because a feature of the conjugate-gradient method is that searching directions previously done basically are never repeated, the second search will be done in a direction perpendicular to this direction. Hence, we could not reach the true minimum point.

Another example with which the program could not do anything, is shown in Fig. 6.1 (b). When the initial position x_0 and the trial position x_1 happen to appear like in this figure, evaluation of the curvature by Eq. (6.1) leads a negative a, and accordingly the parabola is interpreted as upper concave one. Then, there would be no minimum point. In this case, the program proceeds by only restarting from x_1 as the initial point.

Essentially, this case cannot be solved by parabolic approximation. We should find somewhere in which parabolic approximation becomes good by

[1]There are many algorithms for the line minimization, which does not rely on the parabolic approximation. But, these algorithms should evaluate the functional values at many points. If too many time is spent for this evaluation, the merit of the conjugate-gradient method would be totally spoiled.

some ways, such as changing a trial positionx_1

When you encounter this problem, what you can (should) do is check the form of energy curve (adiabatic potential) in this direction. For this purpose, program is slightly modified as follows. In a source file `pw_relax.f`, find subroutine `Atom_Optimize`. At the beginning of this subroutine, you will find a variable

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

Change the variable `step_flag` to be `.TRUE.`. Then, `pwm` stop entering conjugate-gradient process, instead carries out evaluation of energy at equidistance steps in a given direction. The calculation is done in subroutine `Atom_Line_Step`. The unit step width and the total number of steps are defined by `unit_dis` and `u it_max` in this routine. User can change these parameters according to their needs.

The result is shown in an output file `pwm_*.sum`

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

の後、

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

Each line lists the step number, displacement$u$, the total energy $E_{\rm tot}$, and the force component in the searching directionf_s are in order. Because the derivative of $u-E_{\rm tot}$ with respect to u gives f_s, its relation should be checked.

It is wise, anyway, to be around 5 the step number of iteration of conjugate-gradient, (`maxIter0atom`).

6.2 Optimization of cell parameters

Similar to atom optimization, `pwm` can optimize the cell parameters, because stresses can be easily evaluated. Stress tensor σ_{ij} has the same role as Hellmann-Feynman forces do in atom optimization. Correspondingly, the role of displacement is replaced with strain tensor ϵ_{ij}.

6.2.1 Preparation of calculation

As in atom optimization, first execute a SCF calculation for initial cell parameters, with obtaining output wavefunction. Then, rename wavefunction file to `inip_*.wfn`, end edit `pwm.para`

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

First, set `Job Type` to be 2 in order to activate the cell optimization process. In this time, user has two choices whether atom positions are also optimized or not, by setting the next input line.

The maximum allowed number of iterations for cell optimization is set in `maxIter0cell`. By default, conjugate-gradient method is employed also in this case.

`pwm` will stop the conjugate-gradient process if the iteration exceeds `maxIter0cell` or if the residual stress becomes less than `stress_tol` (default value 2×10^{-5} (a.u.)).

When one optimizing calculation does not achieve sufficient convergence, user can continue the optimization by setting `Resume cell relaxation` to be 1. In this case, the data of cell parameters and atom positions achieved in the previous calculation is stored in file `pwm_*.var`, and are read in at the next calculation. Hence, it is not required to rewrite the crystal data file `*.prim`.

6.2.2 Read output files

Here, we show a nontrivial example of cell optimization for $\alpha-$boron. This crystal belongs to the trigonal system with the space group D_{3d}^5. In this case, even under hydrostatic pressures, the strains are not isotropic. Hence, a usual method of isotropic change in volume is not suitable to describe strains under hydrostatic pressures. The present method provides correct description in this case.

The lattice parameters are used from the experimental values as

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

The corresponding stresses are obtained as

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

For quick view of result, it is better to see an output file `pwm_*.sum`. In this file, find a title

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

After this title, the conditions of cell optimization follow as

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

Then, description of conjugate-gradient process with respect to the cell parameters follows

```
IC       Etot0          Etot          del_E          epmin        res stress        its sea comp
 0    -68.47157622    -68.47157622       0.000E+00     0.000E+00      0.109E-02     0.641E+00
 1    -68.47157622    -68.47982412      -0.825E-02     0.248E-01      0.328E-04     0.132E-01
 2    -68.47982412    -68.47983356      -0.944E-05     0.781E-03      0.244E-05    -0.643E-03
 3    -68.47983356    -68.47983356       0.000E+00     0.781E-03      0.244E-05     0.138E-02
   iteration stop
 because the change in E   0.00E+00
      became less than that allowed by RSstress  2.00000E-05

<<<<<<<<< final results <<<<<<<<<<<<<<<<<<<<<<<<<<<<<
 STEP    Etot            delE           resid           rsf            rfm
100     -68.47983356    -0.8715E-08    0.7725E-10    0.5726E-04    0.2039E-03
 Etot= -6.84798336E+01 (Ry/cell)
       -9.31717578E+02 (eV/cell)
```

Here, for simplicity, we extract only relevant portions following a header `IC`. `IC` indicates step number of the conjugate-gradient process with respect to cell optimization. The initial value (`Etot0`) and the final value (`Etot`) of the total energy at each step, change in the total energy (`del_E`), the strain of minimum point (`epmin`), the magnitude of residual stress (`res stress`), and its component in the searching direction (`sea comp`) follow.

As iteration proceeds, you can see that the total energy and the residual stress become small.

As a consequence, the cell parameters become

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

Finally, stress tensor is written out

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

As seen, the diagonal components become small compared with the initial value by two order of magnitude.

6.2.3 Discussion about convergence

Convergence problems in cell optimization is basically the same as those in atom optimization as described in Sec. **??**. If something trouble happens, the best thing is to see what happens with the adiabatic potential along the searching direction.

In source file `pw_cellrelax.f`, find subroutine `Cell_Optimize`. Then, in the header part, find a variable declaration

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

This variable `step_flag` should be changed to `.TRUE.`, then the line step is activated. Actual line step is carried out in subroutine `Cell_StepByStep`. There, you can change default values of step width of cell parameters (`unit_step`) and the number of step(`nstep`) according to your needs.

The result is shown in file `pwm_*.sum`. Find the title

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

Then, look at the following portion

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

For each strain (`e`), the total energy(`Etot`), the stress component in the searching direction (`ssd1`) are listed. Plot `e` versus `Etot`, and `e` versus `ssd1`. The latter gives the derivated of the former.

The accuracy of cell parameters is determined by that of strainϵ_{ij}, and the latter is expressed by the residual stress σ_{ij} as

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

where C_{ijkl} are elastic constants. In evaluating Eq. (6.5), you can replace the elastic constants with bulk modulus.

As an example, let estimate the accuracy of cell parameters of $\alpha-$B. The residual stress is $\Delta\sigma \approx 10^{-6}$ [Ry/Bohr^3]. Assume the bulk modulus B of boron being similar to that of Si ($B = 100$ [GPa] $= 6.8 \times 10^{-3}$ [Ry/Bohr^3]). Then, the accuracy of cell parameters becomes$\Delta\epsilon \approx 1.5\times10^{-4}$, that is 0.01% is concluded.

How to cut plane waves When the lattice parameters are varied, a special caution is required, unlike in atom optimization. Once the cell shape is changed, a way of cutting planewaves becomes different. For a fixed cutoff radius r_c or energy $E_{\rm cut}$, the number of planewaves $N_{\rm pw}$ inside the cutoff sphere may be different. In this case, unphysical discontinuity may appear in the change in the energy.[35] On the other hand, if a set of original planewaves is fixed, and thereby fixed $N_{\rm pw}$, the expansion surface of planewaves in G space is no longer sphere, and thereby resulted in improper cutting. This problem becomes especially troublesome for strong anisotropic crystals.

This contradiction forces us to compromise at some points. In `pwm`, the latter method is employed, in order to avoid unphysical discontinuity. But, users should remind that calculation may be correct only for small strains. Qualitatively, what to extent this approximation is valid of course depends on the conditions of calculation and materials, but roughly speaking, strains more than 10% is worth to suspect the correctness.

6.2.4 Pressure dependence

`pwm` can optimize cell parameters under hydrostatic pressures. In `pwm.para`, an option is added as

```
pressure_GPa=
  10.0
```

The applied pressure is input in GPa. Positive value means compression, while negative value expansion.

In `pwm`, the conjugate-gradient process is repeated until the internal stress is balanced with the external pressure.

6.2.5 Related options

- steepest_atom_relax ON

- steepest_cell_relax ON

In atom and cell optimization, by default the conjugate-gradient method is applied. But, the conjugate-gradient method is not always best one, as described in Sec 6.1.4. In some times, the steepest-descent method is more favorable. In this case, the above options let the process do so.

- pressure_GPa=
  1
  Specify an externally applied pressure in GPa.

- trial_step0atps=
  0.001
  In atom optimization, specify the trial displacement u_1 in the line minimization. The atomic unit is assumed.

- trial_step0cell=
  0.001
  In cell optimization, specify the trial strain ε_1 in the line minimization.

Chapter 7

Electronic band structure

After completion of SCF calculation, band and DOS calculations are possible. Prior to proceeding to these calculation, knowledge of Brillouin zone and other related topics are prerequisite. For example, user should know that the Brillouin zone of FCC is as shown in Fig. 7.1. Brillouin zones for all Bravais lattices are tabulated in Ref. [24]. If you want to draw Brillouin zone by yourself, Mathematica's notebook `MakeBZ.nb` gives users a way to do so.

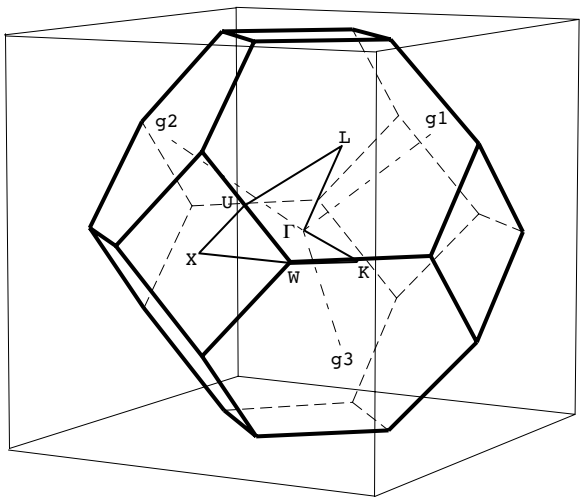

Figure 7.1: Brillouin zone of FCC

Band and DOS calculations are done in `pwbcd`. File dependence related this program is shown in Fig. 7.2. Program `ayband` and `pdosdr` are

drawing programs into PostScript file, which are written by Professor A. Yanase. First of all, data of charge density of SCF calculation `pwm_*.rho` is required. Although k point sampling data `inip_*.kpt` is not necessary, file `inip_*.inp` is still needed, because this file keeps the type of potentials along with other conditions. For both of band and DOS calculations, the same input file `bcd.para`, but its description is different for different calculations. Maybe, you may be better prepare different input format `band.para` and `dos.para` for respective calculations, and then rename one of then to `bcd.para` depending on which calculation is chosen.

7.1 DOS structure

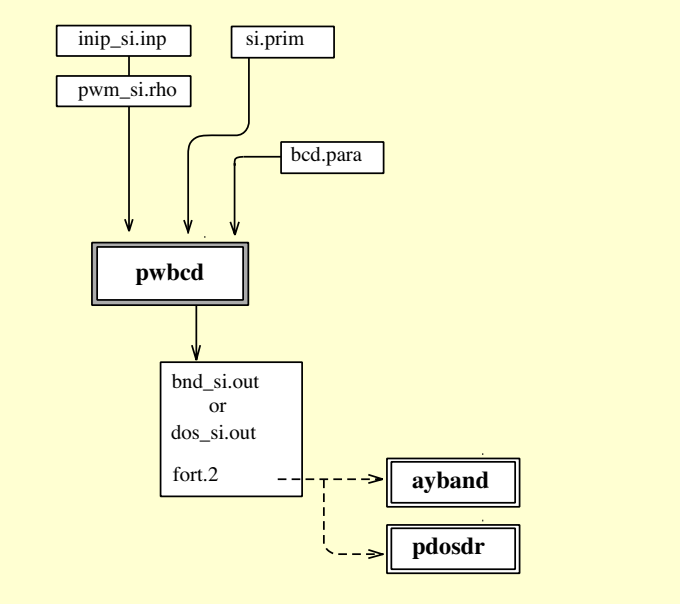

Figure 7.2: flow of `pwbcd`

7.1.1 Preparation

For DOS calculation, input file `bcd.para` looks like

```
JobType
dos
Input file name
```

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

In the following, meaning of the parameters is explained.

`JobType`	Job type = dos
`xtl.name`	crystal name
`nkdiv`	division number of k line
`NBUP`	number of bands drawn
`iscan`	options for debug, 1: default
`ilp`	print option, 1: default
`isymm`	options for symmetrization, 1: on, 0: off
`ienun`	energy unit, 0: Ry, 1: eV

The most important parameter is the number of division of the Brillouin zone. This is done by specifying `nkdiv`. In each axis of the parallelepiped of the first zone is divided by `nkdiv`. The number of bands drawn (`NBUP`) should be less than `NHDIM`. Usually, about twice of the number of valence bands is sufficient. But, for simple metals such as alkali metal, the number of valence bands is too small (only one or two), so that further inclusion of conduction bands is desirable.

Other parameters are only for debug purpose, and hence usually the above setting is enough.

`pwbcd` prints out the result in file `dos_*.tbl` and `fort.2`. In order to display these data, a Mathematica notebook `DOSshow.nb` is ready for doing conventionally. In this case, raw data in file `dos_*.tbl`. For further processing, data file `fort.2` containing symmetry-decomposed spectra can be displayed by `pdosdr`. This utility is created by Prof. Yanase.

In DOS calculation, the k sampling is taken so that the parallelopiped spanned by three primitive vectors, $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, are uniformly segmented by `nkdiv`. For example, in case `nkdiv`=4, four k points are taken in

the $\mathbf{G}_1$ direction, as

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

in G_1 units. Unlike k sampling method by Monkhorst-Pack, it is noted that the sampling point starts from the zone center. Among them, those point which are outside the first zone are automatically held back into the first zone. Points which can be transformed by symmetry operations are eliminated from a set of sampling points. This means that the parallolopiped is reduced to the irreducible wedge. Accordingly, weight factor is associated with these k points.

`pwbcd` diagonalizes Hamiltonian at each k point, and sort the eigenvalues in ascending order. When an option is set as `isymm=1`, the set of planewaves are symmetrized before diagonalization.

At the last of calculation, all the valence electrons are assigned each bands from the lowest bands, in order to obtain Fermi level E_{F}.

SGI

On SGI machines, when compiling `pwbcd`, it is recommended that no higher optimized level than `-O` is applied. This is due to subtle compatibility of TSPACE. Whenever TSPACE is linked, it is more safe to use a compiler option `-static`.

7.1.2 Results of DOS calculation

Results of DOS calculation are written out in file `dos_*.tbl` and `fort.2`. Prior to seeing these results, the calculation conditions should be checked in a file `dos_*.out`.

After the title

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

the calculation conditions are listed. Then, a subtitle appears as

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

which follows information about k point sampling. In the above example, k points are initially created by eight. Then, symmetry reduces the number of k points to the next three,

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

In this list, the name of k point, the coordinates in the conventional, and in the primitive base appear, which are followed by the magnitude (`DK`), the number of the star, (`Nstr`), the k point wight (`WTK`), the index as to whether inside of the zone or outside (refer TSPACE).

After that, it describes processes of symmetrization of planewaves and diagonalization of Hamiltonian for each k point. Codes of Osaka2k such as `pwm` are basically described on the primitive base, while TSPACE is described on the conventional base. For this reason, `pwbcd` heavily does conversion coordinates between these expressions, and checks are made many times. These checks are recorded in file `dos_*.out`. Usual users do not need this description.

`dos_*.out` concludes with estimating Fermi level $E_{\rm F}$.

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

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

Nel=      8.0000

Ef=      6.2896
```

Within the resolution of a given mesh, the energy gap is also estimated. The above example says that the energy gap is 0.8889 eV, which is difference between the valence top at (000) and the conduction bottom at (220)/4.

After subtitle `final distribution of the valence bands`, occupations of the valence top bands are displayed over the zone. For Si, because it has four valence bands, the forth band is the topmost occupied band, unless there is no degeneracy. When degeneracy takes place, all the degenerate bands are considered as the valence top level. The band index of these

degenerated top bands are kept by indeces from `IP2` to `IP3`. `E2` and `E3` are respective energies. In the present case, the degeneracy is rigorous for symmetry reason. But, even without symmetry reason, those levels which fall within the energy resolution `ERESO` are regarded as being degenerate. In this case, those bands from `IP2` to `IP3` are equally occupied. `pwbcd` checks if the occupation obtained in this way is consistent with the total number of electron `Nel`.

7.1.3 Display

For displaying DOS structure, there are two choices.

(i) Line spectrum

The most primitive method to display DOS spectra is simply plot the raw data in `dos_*.tbl`. The format of this file is

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

The first column lists index of k point, the second does the KS levels, and the third does the weight of the k point. Mathematica's notebook `DOSshow.nb` reads this file, put these data in Gaussian distribution function, and plot that. An example obtained in this way is shown in Fig. 7.3 (a).

When using corse mesh points, you see spurious peaks. If you want to look for van Hove singularity, this is really trouble unless a huge number of k mesh point is taken.[1]

(ii) tetrahedron method

The tetrahedron method is a sophisticated algorithm of interpolation for smoozing discritized DOS curve. [36]

`pdosdr` do that by reading an output `fort.2` of `pwbcd`, when `JobType = dos`. The format of `fort.2` will be explained in Sec. 7.2.2. Because `pdosdr` is originally created for other use, the way of description for lattice is different from Osaka2k; `pwbcd` uses the primitive base, while `pdosdr` uses

[1] This point is leaned by Prof. Yanase.

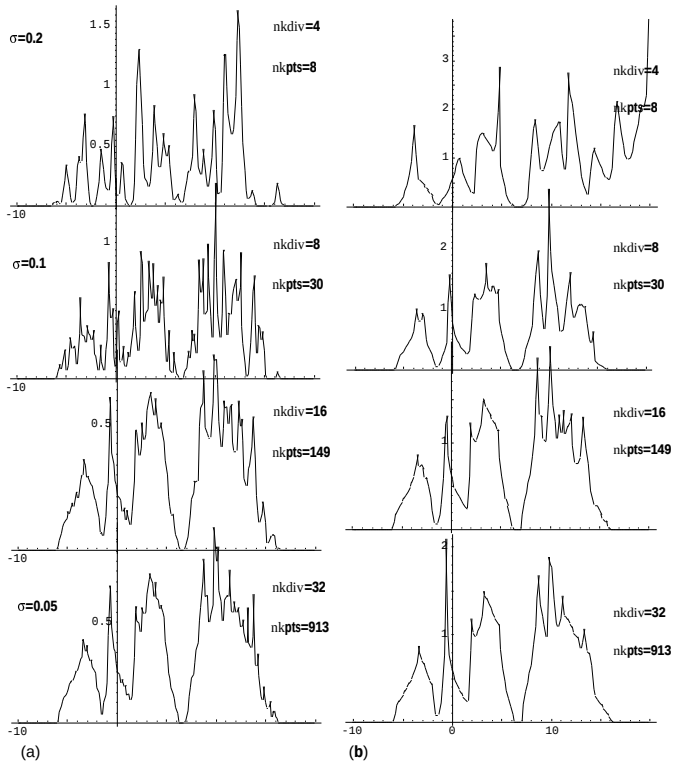

Figure 7.3: DOS of Si. (a) line spectra broaden by Gaussian, (b) spectra drawn by the tetrahedron method

the conventional base. For a case such as FCC, user are required to bridge these two expression in k sampling. [2]

For Si case, for example, the Brillouin zone is divided by $2 \times 2 \times 2$ in `pwbcd`. Because $\{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3\}$ is not orthogonal system, the above mesh yields (111)/2 as well as (000) and (001) in the conventional base ($2\pi/a$ unit). On the other hand, `pdosdr` divides uniformly the space $\{\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*\}$, because of use of conventional base. If we set the division number two, extra points such as (100)/2, (110)/2, in addition to the above k points. As to these extra points, because `pwbcd` does not sample, this setting causes error in `pdosdr`.

Hence, in `pdosdr`, the division number should be 1. Then, `pdosdr` takes

[2]The authors did not check completely for all crystal systems, although he believes correctness of the present treatment.

four k points, i.e., (000)/1, (100)/1, (110)/1, (111)/1. Because (111)/1 is equivalent to (000)/1, and (110)/1 is to (100)/1, among three k points in `pwbcd`, only two are used in `pdosdr`. The other is discarded. This waste of computation cannot be helped.

Now, we explain how to use `pdosdr`. Prepare an input file `pdos.para`, as

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

The first line is related to magnetic states, and in the present case, simply follow the above example. The next three numbers indicate the division numbers in the k space, `NX`, `NY`, and `NZ`. Care must be paid in order to keep consistency the division numbers in `pwbcd`, as described above. The next line is set to be zero, which is used for decomposition of DOS into atomic orbitals. The following three numbers are the lower limit of energy axis (`ESTART`), the upper limit (`ENEND`, the width of energy step (`DE`). The mesh points of the energy axis are therefore `ENEND-ESTART)/DE+1` The last number is the number of electrons in the primitive unit cell

An output file `fort.12` lists as

```
SPIN  1
  1  -10.0000    0.000   0.0000
  2   -9.8000    0.000   0.0000

150   19.8000    1.689  27.1343
151   20.0000    1.438  27.4481
electron number=  8   EF=  6.80000000000000
```

The spectrum data are listed between the first and last comment lines. Each line lists the index of energy mesh, energy, DOS, and its convolution up to this point.

This data can be displayed by Mathematica's notebook `PDOSshow.nb`. By removing the first and last lines of `fort.12`, it is read by `PDOSshow.nb`. In Fig. 7.3 (b), an example is shown. Even when mesh points are small, DOS curve looks smooth. But, if you want to obtain van Hove singularities, it is invariably true that you have to take a large number of mesh points.

7.1.4 Fermi surface

For metals, Fermi surfaces can be drawn by using output of `pwbcd`. As shown in Fig. 7.1, `pwbcd` outputs file `dos_*.sum`, in which bands are written from k to k, disregarding information of irreducible representations as in `fort.2`. This data are transformed to Mathematica's notebook `MakeFE.nb`, where Fermi surface is constructed. The Fermi level is given in `dos_*.out`. Actually, this notebook regards the Fermi level simply as an energy level, so that any value is acceptable. This means that this notebook can be also used to draw equi-energy surface.

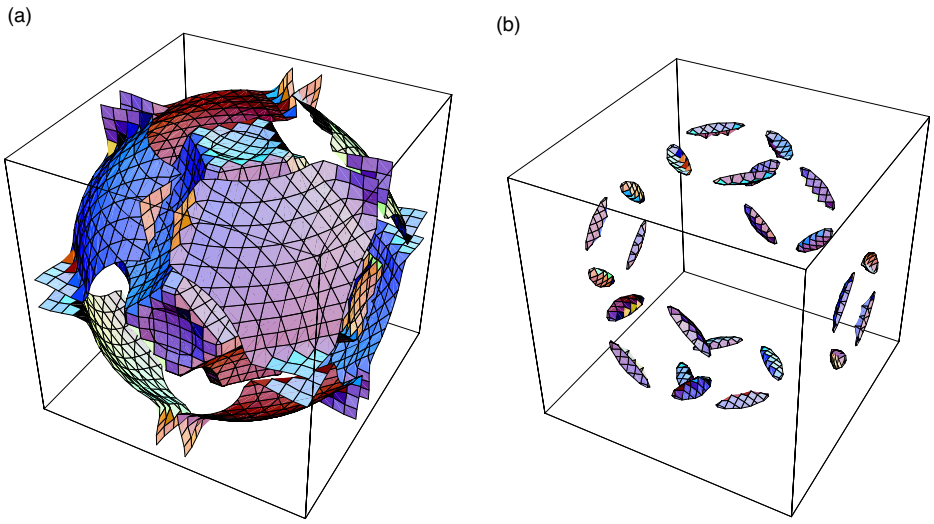

Figure 7.4: Fermi surface of Al: a) 2nd band, (b) 3rd band

In Fig. 7.4, an example of Fermi surface for Al is displayed. Here, the form of extended zone is used. To draw such figures, a computation time becomes sometimes a hour. Be patient! You may further want to superimpose the boundary lines of Brillouin zone such as in Fig. 7.1 to Fermi surface. `MakeFE.nb` does not do so. TSPACE provides utilities for this purpose, which are more flexible. But, in this case, you should understand underlying principles for contour mapping, along with many other techniques.

7.2 Band structure

The band calculation in `pwbcd` calculates bands at k points along arbitrary lines, although the lines are usually chosen to be symmetry lines. These lines should be singly connected. As in DOS case, two ways for display the result are ready. One is simply plot each bands without information of symmetry, by reading file `band_*.tbl`. The other is sophisticated method of using `ayband`, which interpolates and extrapolates points by using symmetry properties fully. Outputs of `ayband` is PostScript file. Then, the file can be handled by many drawing programs, such as Illustrator. `ayband` is created by Prof. A. Yanase.

7.2.1 Preparation

For band calculation, input file `bcd.para` looks like

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

In the following, meaning of each parameter is described.

The most important input is specification of lines, on which band dispersions are drawn. `pwbcd` assumes that the drawing lines are singly connected. Hence, the drawing lines are expressed by sequence of node points k as

$$A \to B \to C \ \cdots$$

The number of nodes is given by `NKPTS` in `bcd.para`. Then, each node point k, `KB(3)/ICB`, is given in primitive base.

`JobType`	Job type = bnd
`xtl.name`	crystal name
`NKPTS`	number of lines drawn
`KB(3), ICB`	terminal k points of each line
`NDIV`	division number for each line
`NBUP`	number of bands drawn
`iscan`	options for debug 1: default
`ilp`	print option 1: default
`isymm`	option for symmetrization 1: on, 0: off

In the above example, five symmetry lines, $L-\Gamma$, $\Gamma-X$, $X-W$, $W-K$, $K-\Gamma$, are taken. [3] The segment number of each line is given by `NDIV`.

The number of bands which you want to draw is specified by `NBUP`. `pwbcd` calculates bands from the lowest up to `NBUP` at each k point. As in DOS, it may be as large as the number of electrons in the primitive unit cell. But, if too small, it should be more than that. When setting `use symmetry = 1`, this caution is particularly important. In this case, `pwbcd` enumerates bands at *each symmetry block* by `NBUP`. If `NBUP` is too small, it may happen for some band on a symmetry line to miss to connect to that on another symmetry line. In this case, `NBUP` should be increased. The remaining parameters are for debug purpose, so usually there are left as it.

7.2.2 Result of band calculation

`pwbcd` calcultes bands along given lines, and yields output file `bnd_*.tbl` and `fort.2`. Before seeing result, users should check if the calculation conditions are really what you intend. This can be seen in file `bnd_*.out`, as in DOS.

For quick view band diagram, data `bnd_*.tbl` is transformed to a 2-dimensional plotting program. The format of `bnd_*.tbl` looks like

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

In each line, first the index of k point comes, then an energy level comes. Therefore, plotting it is easy. An example obtained in this way is shown in

[3] Prof. A. Yanase points out a wise selection $\Gamma \to K \to X(1,1,0) \to W \to L \to \Gamma \to X(1,0,0)$

Fig. 7.5. The abscissa represents k lines in order $L-\Gamma-X-W-K-\Gamma$, and each line is segmented by 5. Due to lack of symmetry information, connectivity between adjacent levels has to been decided manually. In this example, this determination is not difficult, but for more general case, ambiguity always is associated with. At the zone boundary, sometimes the asymptotic behavior of dispersion of degenerate bands is important. In this case, symmetry properties help us greatly to remove such ambiguities. But, full use of symmetry properties are highly skillful work, so that user should lean deeply group-theoretical methods.

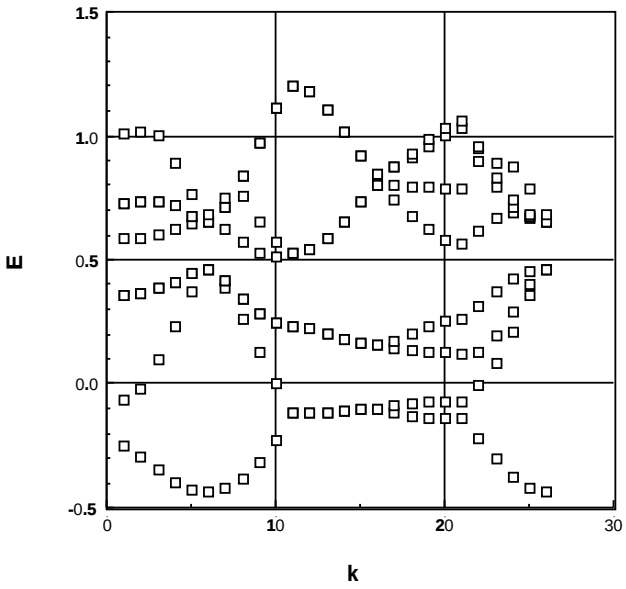

Figure 7.5: Primitive plotting of bands of Si. The output of `pwbcd` is simply plotted without use fo symmetry properties.

`ayband` does this in place of you. `ayband` connects relevant points smoothly by spline curves. To which point should be connected to which is judged by using irreducible representations. Also, for curvature at the zone boundary, `ayband` checks if k dependence is linear in k or quadric, by using cumbersome processes which are seemingly redundant calculations.

> Two bands which belong to the same irreducible presentation do not cross, *i.e., no closing rule*. Accordingly, there is no *accidental* degeneracy. `ayband` is strict with this principle. I have no compliant with this treatment in implementation, because actual code should have always definite criteria, otherwise we could not write code. However, I am not sure if this is in principle correct. For example, C. Herring discussed various properties of accidental degeneracy *if* the accidental degeneracy presents.[37] I am not sure if accidental degeneracy without any symmetry reason can happen.

Output file `fort.2` is a complete data set in terms of symmetry properties. This file is expected to put in `ayband`.

The data format of `fort.2` is as follows.

Data are gathered according to k points, while in each k point data are gathered according to the irreducible representation of the small group of k. The minimum data block looks like

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

The first number is a serial number, which follows the next four represent k in presentation `KB(3)/IC`. In this examle, $(5, 5, 5)/10$, that is, L point. User should note that k is presented in the conventional base. Next comes a spin-related parameter (`IUD`), which follows the name of k point.

The remaining four are the index of the irreducible representation (`MRN`), the dimensionality of the representation (`MWET`), the number of star k (`NST`), and the number of eigenstates belonging this block (`NEIG`).

After carried return, `NEIG` eigenvalues are enumerated in ascending order in energy.

`ayband` uses information involved in `fort2`.

In order to execute `ayband`, further input data are required. `fort.3` is another input file. There contains parameters to control drawing process. The format looks like

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

```
  6  6  0  8   0  0  0  8 SM
0.5                      Fermi level
Si PseudoPotential
```

The meaning of these parameters are described in the following.

MAGNET Magnetic state. `pwm` uses option `NONMSPIN-ORBIT` only.

NLCOMP, NSPIN,IFILE the number of DOS components with respect to angular-momentum, spin state. Set both to be 0. `IFILE` is the file index of the output file.

IPR, JMARK,IPOINT The first two are parameters controlling output. By default, set as in the above. `IPOINT` is the size of characters for the representation in the band diagram. item [E0, EM, YM, XM] Minimum and maximum of energy of plotting, and the scale of $x-$ and $y-$ axis in mm.

NAXM the number of lines in band diagram. Individual lines are described in the following.

line description Specify symmetry lines by giving two terminal points $\mathbf{k}_1$ and $\mathbf{k}_2$. $\mathbf{k}$ is given three numerators and the common denominator. This description appears for `NAXM` lines. The name of terminal points are added for users convenience only, and are not read in `ayband`.

EF Fermi level

TITLE The title of figure

In this way `ayband` requires again line specifications, independent of those of `pwbcd`.

In Fig. 7.6, a band diagram obtained in this way is displayed.

You can see that dispersion curves are smoothly drawn, and connectivity at the zone boundary is made clear.

In the figure, the numbers indicate the label of irreducible representation. This numbering is in accordance with `TSPACE`, the way of which is its own one. Users who have well read reference may want to use more orthodox labeling. In this case, user are expected to examine correspondence of labeling between `TSPACE` and the reference you want.

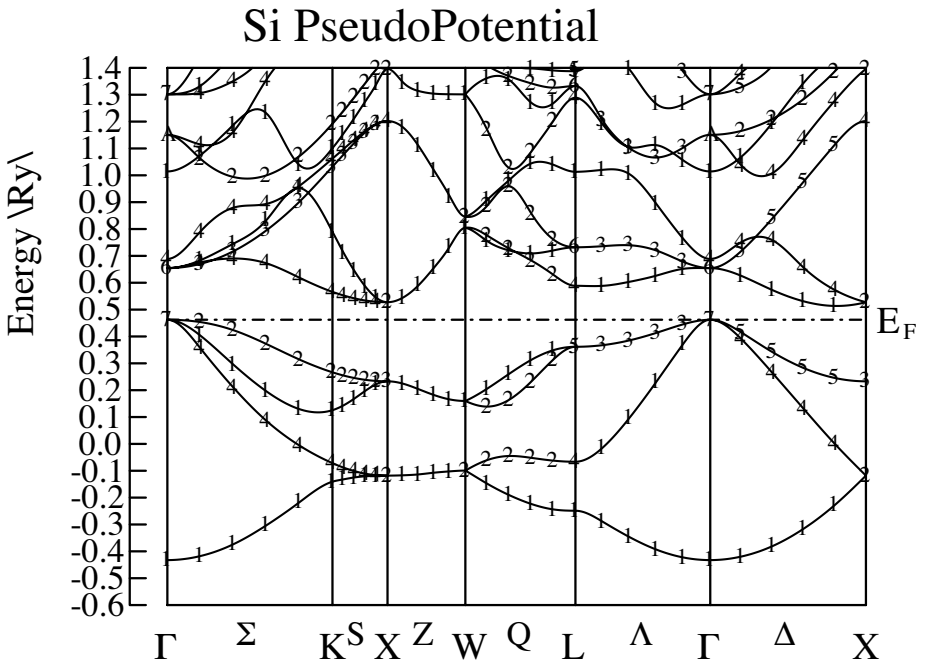

Figure 7.6: band diagram of Si drawn by ayband. The line lengths of different k points are given by the correct length in k space

> As to time-reversal symmetry, a note should be added. When time-reversal symmetry yields extra degeneracy between different irreducible representations, `ayband` displays only one of them. For example, a zincblend crystal (S.G. T_d^2) such as GaAs has extra degeneracy between Δ_3 and Δ_4 on Δ line, due to time-reversal symmetry. `ayband` attributes only 3 as the label. Therefore, when you enumerate bands and degeneracy, you should count by taking another invisible one into account.

A troublesome point of character table in the literature is that there are actually two ways of presentations; normal irreducible representation (vector presentation) and ray presentation (projective or multiplier representation). [4] `TSPACE` uses ray presentation. In my impression, vector representation is more frequently used in the literature of band calculations. Maybe, Koster

[4] An example of the vector representation is Koster[38]. This representation is easy to understand, but actual character table is apt to be big. Enumeration of this representation

table [38] is one of the most frequently cited one. Therefore, the correspondence of representations between TSPACE and literature such as Koster, BSW, etc.

TSPACE can print out complete list of irreducible tables by calling TSIRDS. TSPACE uses ray representation $D(\alpha)$. It is an irreducible representation $\Gamma(\alpha)$ of the small group k multiplied by a phase factor

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

where τ_α is a nonprimitive translation associated with rotation α This representation $D(\alpha)$ looks neat because superfitious factor associated with nonprimitive translation is removed, and thereby has definite correspondence to the point group except zone boundary.

Let us see an example of nonsymmorphic space group, say, Si. On Δ line, subroutine TSIRDS prints out character tables, as

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

in file fort.15. In this table, all the matrix element for all the symmetry elements are listed (In the present, only diagonal elements are extracted for clarity). Meaning of the index of symmetry elements and index of representation should be referred to in TSPACE [24]. The character of this ray presentation is compiled to show in the left of Table 7.1.

By seeing this, you can find definite correspondence of character table between the present $D(\alpha)$ and that of point group C_{4v}. On the other hand, in the vector representation $\Gamma(\alpha)$, complex number η appear. In the limit

for all 230 space groups is tremendous task. On the contrary, ray representation achieves greatly economy of table. Hence, Kovalev[39], Bradley Cracknell [40] thoroughly describes character tables which cover all the space group. Ray representation is, on the other hand, higher skill to understand.

Table 7.1: Character table on the Δ line. The left shows multiplier representations D, while the right shows vector representations Γ. The second row lists non-primitive translation $\tau = (1/4, 1/4, 1/4)$ associated with rotation. The first column of the left part indicates the label in TSPACE.

	D	E	C_2	$2C_4$	$2\sigma_{\rm v}$	$2\sigma_{\rm d}$
				τ	τ	
1	Δ_1	1	1	1	1	1
2	Δ_1'	1	1	1	-1	-1
3	Δ_2	1	1	-1	1	-1
4	Δ_2'	1	1	-1	-1	1
5	Δ_5	2	-2	0	0	0

Γ	E	C_2	$2C_4$	$2\sigma_{\rm v}$	$2\sigma_{\rm d}$
			τ	τ	
Δ_1	1	1	η	η	1
Δ_1'	1	1	η	$-\eta$	-1
Δ_2	1	1	$-\eta$	η	-1
Δ_2'	1	1	$-\eta$	$-\eta$	1
Δ_5	2	-2	0	0	0
	$\eta = \exp[-i\mathbf{k}\cdot\tau]$				

$k \to 0$, this factor becomes 1, so that you can still find correspondence to the presentation of the point group C_{4v}.

But, at the zone boundary, it is not easy. When the dimension of a representation is one, the representation becomes equivalent to that of point group, but complex numbers appear in the character. When the dimension is more than one, correspondence to that of point group disappears. Probably, irreducible representations of X point is the most complicated one. The ray presentation looks like,

```
IMR NO  1  DIMENSION=  2
    1 2 3 41618192225262728404243 46
1 1 + + + + 0 0 0 0 0 0 0 0 - - - -
2 2 + + - - 0 0 0 0 0 0 0 0 - - + +
IMR NO  2  DIMENSION=  2
    1 2 3 41618192225262728404243 46
1 1 + + + + 0 0 0 0 0 0 0 0 + + + +
2 2 + + - - 0 0 0 0 0 0 0 0 + + - -
IMR NO  3  DIMENSION=  2
    1 2 3 41618192225262728404243 46
1 1 + - + - J I J I 0 0 0 0 0 0 0 0
2 2 + - - + J I I J 0 0 0 0 0 0 0 0
IMR NO  4  DIMENSION=  2
    1 2 3 41618192225262728404243 46
1 1 + - + - I J I J 0 0 0 0 0 0 0 0
2 2 + - - + I J J I 0 0 0 0 0 0 0 0
```

As before, it is arranged as the character table in the left of Table 7.2.

As Koster table [38] lists only vector representations (the right of Table 7.2), the characters converted to the vector representation $\Gamma(\alpha)$ by Eq. (7.1) are written in output file `fort.18` of `pwbcd`. There, character tables are listed at each k point.

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

Table 7.2: Character table of X point. The left shows multiplier representations D, while the right shows vector representations Γ. For simplicity, symmetry elements whose character vanishes are omitted from this table. The first column of the left part indicates the label in TSPACE.

	D	E	C_{2x}	$2C_2'$	$2\sigma_\mathrm{d}$
			τ		
2	X_1	2	2	0	2
1	X_2	2	2	0	-2
4	X_3	2	-2	$2i$	0
3	X_4	2	-2	$-2i$	0

Γ	E	C_{2x}	$2C_2'$	$2\sigma_\mathrm{d}$
		τ		
X_1	2	2	0	2
X_2	2	2	0	-2
X_3	2	-2	2	0
X_4	2	-2	-2	0

```
Characters
#IR=  1     ND=  2     MG= 16
JG=         1     2     3     4    16    18    19    22    25    26    27    28
  Re=    2.00  2.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
  Im=    0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
JG=        40    42    43    46
  Re=   -2.00 -2.00  0.00  0.00
  Im=    0.00  0.00  0.00  0.00
```

Characters are listed with decomposing to its real and imaginary parts for each symmetry element, where `JG` is its index. Now you can directly compare the present and the literature labeling. Although this task is the one, which requires you patience and painful effort, it will provide precious experience.

7.3 Display of wavefunctions

In some times, you may want to draw wavefunctions in the real space. Osaka2k provides two ways to do so. These are described in Ref. [42].

Appendix A

Graphics display of a crystal by Mathematica

A.1 coordinates systems

Most of graphics data of Osaka2k can be handled by Mathematica's notebooks. Among these nobebooks, only `ChgDnst.nb` is described in this appendix. Because projecting of charge density distribution on a cross section is frequently used, user should be familiar to handle the coordinate system in a general way. Nowadays, there are many commercial programs are available to display such a distribution. Nevertheless, you cannot use it with foolproof. I mean that without knowledge of coordinate system and of how surfaces are expressed, you cannot obtain correct answer. Efforts for understanding how express nonorthogonal system and how specify a cut plane in a particular program, even it is a well documented commercial program, amounts to almost the same effort to create the same program. You must know what you are doing with a specific coordinate.

Here, as nontrivial example, we take α oxigen crystal, which belongs to monoclinic crystal. The space group is C2/m, and base-center (C-center), and the crystal structure is shown in A.1 (a).

Basically, Mathematica's 3-dimensional plotting routines assume to use equi-spacing orthogonal mesh. Owing this, tedious conversion of coordinates are required for non-orthogonal systems. Although user are not required to know details of coordinate transformation, he must know how specify a cut plane he wants to draw.

In `ChgDnst.nb`, by section 2. (`Construct crystal structure`), a given

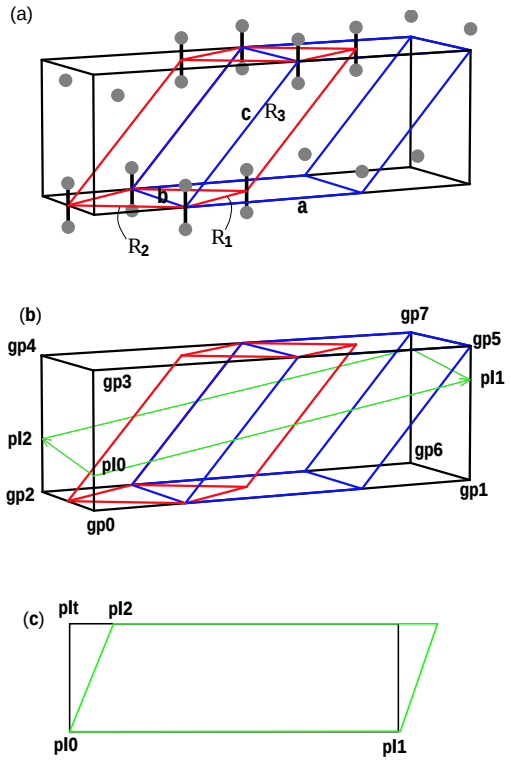

Figure A.1: Cutting a cross section out of the crystal. (a) $\alpha-$ oxigen crysal. The primitive unit cell (red) and the conventional unit cell (blue) are indicated. Graphics frame covers both of unit cells. (b) a cut plane (green) is specified by three points, {`pl0`, `pl1`, `pl2`}. (c) 2-dimensional charge density is plotted on the rectangular spanned by {`pl0`, `pl1`, `plt`}

crystal is described in the folloing, First, in section `read *.prim`, data of the crystal structure is read by opening data file

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

Hence, you should input the name of a crystal file.

The length unit is given by the maximum of $\{a_0, b_0, c_0\}$ (`amax` in Å)

- The primitive unit cell is given by three primitive translational vectors $\{t_1, t_2, t_3\}$ ({ `t1, t2, t3`} in the notebook)
- The conventional unit cell is given by three translational vectors $\{t_{c1}, t_{c2}, t_{c3}\}$ ({ `tc1, tc2, tc3`})

- Define a global region so that it includes all these unit cells. The global region is spanned by three vectors $\{gp_1, gp_2, gp_3\}$ ({ `gp1, gp2, gp3`}), and these are necessarily an orthogonal set.

Given these basic data, `ChgDnst.nb` draws the primitive and conventional unit cells, and cut the global region, as shown in Fig. A.1 (a). The global region is a parallelepiped, whose vertices are termed as $gp0$, ..., $gp7$ (in the figure (b)). In what follows, all coordinates are expressed on this global coordinates.

In this coordinate system, atom positions are expressed as follows.

- The basis are composed of $N_{\rm atom}$ (`natom`) atoms. The positions are given by `ba[i]`.

- Lattices are specified by a set of three integers. This is done in the notebook by calling function `lattice`

```
lattice[l_,m_,n_] := N[l t1+m t2+n t3]
```

  In convenience, several of these are identified by a serial number, as

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

  and

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

- Atom positions are given by calling function `atomPos`

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

This is a list of major variables in the notebook.

Another complication can occur for the present case of oxigen crystal. When the graphics region is clipped by the global region, *gp*0, ..., *gp*7, a molecule O_2 is halfway cut. Accordingly, the molecular unit is hardly seen. Therefore, user should expand the global region further. This is done manually, by following the algorithm of the notebook.

After that, in section `read data file`, input the file name of charge density, as

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

Then, charge density data are read in. After the title, `density statistics`

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

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

`ChgDnst.nb` calculates the maximum and minimum of charge density.

3-dimensional contour map of charge density is plotted by scanning data points inside the global parallelepiped region. User achieves it only by throwing desired parameters in plot command

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

Here, `Contours` is the value of contour at which contour map is created. Hence, this value must be fallen into a range between the mimimum and maximam of charge density, estimated before. Fig. 5.4 is an example obtained in this way in a global region, `gp0, ... , gp7`.

A 2-dimensional cut plane is specified by two vectors. These are specified by three points, `pl0, pl1, pl2`. It is recommended that these three points are expressed by using `gp0, ... , gp7` instead of using absolute coordinates. If you want to take `pl0` a quarter of `gp0-gp3`, define it as

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

The right-hand side contains `gp0, ... , gp7` only.

Because the region spanned by the vectors is, in general, not rectangular, a new vector `plt` is taken as normal to a line `pl0-pl1` as shown in the figure

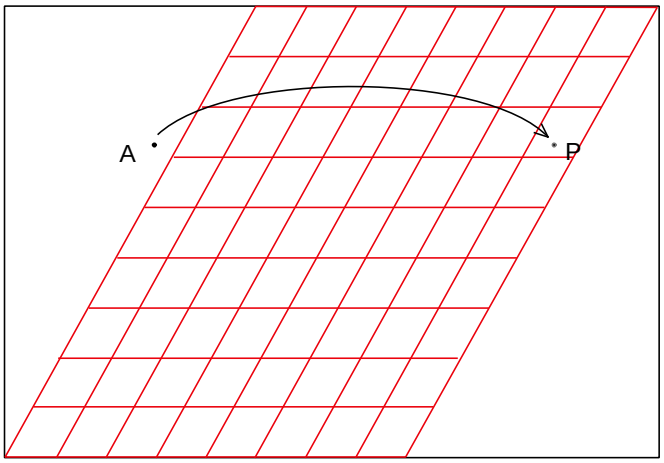

Figure A.2: Coordinate transformation in displaying charge density plot. The primitive unit cell is indicated by red lines, while black frames indicate the global region. Charge density is given at each lattice point. Point A given in the cartesian coordinates is draw back to an equivalent point P in the primitive unit cell.

(c). The black area if the final region on which contour map is drawn. In `ChgDnst.nb`, intersections of the unit cell and atom positions, if any, are drawn as well. An example is shown in Fig. 5.5.

A.2 Charge density contour map

The charge density data in the real space, `rod_r`, are given on the primitive unit cell only, because it is a minimum set. The primitive unit cell is divided by {`na1, na2, na3`} , and the charge density is given only those *lattice points*. The normalized factor is taken to satisfy

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

where N_{el} is the number of electrons in the primitive unit cell

After charge density data are read in the notebook, the charge density is transacted. There are two points to be considered.

First, it is almost always better to expand charge density data on the cartesian lattice points in order to proceed with graphics processes. On the other hand, the original primitive cell on which the lattice points are expanded is in general not an orthogonal system. Even cubic crystals, FCC and BCC are not orthogonal systems.

Another point is that, because the original data are given only inside the primitive unit cell, in order to obtain the charge density at a general point in the whole space, the general point is hold back into the primitive unit cell.

By these requirements, conversion of the coordinates between the cartesian and the original primitive unit cell space is needed. These relation is shown in Fig. A.2.

This conversion is carried out by function

```
fd[l_,m_,n_]
```

in notebook. This function is calculate the charge density at a general point on the whole space. A point `A` on which you want to know the charge density is given by the cartesian coordinate `l, m, n`. This point is draw back to point `P` in the primitive unit cell as shown in Fig. A.2. The primitive unit cell is originally divided by {`na1, na2, na3`} . Because the point `P` is generally not a *lattice point* in the primitive unit cell, the charge density of `P` is interpolated by those of the nearest neighbor lattice points. The interpolated value is return by function `fd`.

All the contour maps of charge density are drawn through calling `fd`.

Acknowledgment

In the course of developing Osaka2k, contribution of pioners has been described in Introduction. Here, I like express my gratitude to other people, mainly students of our laboratory. Particular, H. Fujita and Y. Yamazaki contributed at the initial stage of the project, which is usually the most difficult time. Dr. Dinh van An helped me to translate the part of theory in this manual.

References

[1] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964)

[2] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965)

[3] J. Callaway and N. H. March, *Solid Stete Physics* **38**, (1984) p. 135

[4] R. O. Jones and O. Gunnarsson, Rev. Mod. Phys. **61**, 689 (1989)

[5] *Adv. in Quantum Chemistry* **21**, S. B. Trickey ed., (Academic, San Diego, 1989)

[6] D. M. Ceperley and B. Alder, Phys. Rev. Lett. **45**, 566 (1980)

[7] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981)

[8] V. Heine, *Solid Stete Physics* **24**, (1970) p. 1, and all the volume.

[9] D. R. Hamann, M. Schlüter, and C. Chiang, Phys. Rev. Lett. **43**, 1494 (1979)

[10] N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991)

[11] L. Kleinman and D. M. Bylander, Phys. Rev. Lett. **48**, 1425 (1982)

[12] G. B. Bachelet, D. R. Hamann, and M Schlüter, Phys. Rev. B **26**, 4199 (1982)

[13] N. W. Aschcroft and N. D. Mermin, "Solid State Physics", (Holt-Saunders, Philadelphia, 1976)

[14] J. Ihm, A. Zunger, and M. L. Cohen, J. Phys. C: Solid State Phys. **12**, 4409 (1979)

[15] M. Born and K. Huang, "Dynamical Theory of Crystal Lattices", (Oxford University Press, Oxford, 1954)

[16] H. J. Monkhorst and J. D. Pack, Phys. Rev. B **13**, 5188 (1976)

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. R. Flannery, "Numeric Recipes", 2nd ed. (Cambridge, Camridge, 1992)

[18] M .P .Teter, M. C. .Payne, and D. C. Allan, Phys. Rev. B **40**, 12255 (1989)

[19] M. C. .Payne, M .P .Teter, D. C. Allan, T .A .Arias, and J .D Joannopoulos, Rev. Mod. Phys. **64**, 1045 (1992)

[20] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985)

[21] R. P. Feynman, Phys. Rev. **56**, 340 (1939)

[22] O. H. Nielsen and R. M. Martin, Phys. Rev. B **32**, 3780 (1985)

[23] O. H. Nielsen and R. M. Martin, Phys. Rev. B **32**, 3792 (1985)

[24] A. Yanase, Fortran Program for Space Group (TSPACE), (Shokabo, Tokyo, 1995)

[25] T. Hahn, ed. "International Tables for Crystallography", Vol. A (Kluwer Academic Pub., Dordrecht, 1996)

[26] G. Burns and A. M. Glazer, "Space Groups for Solid State Scientists", 2nd. ed., (Academic, San Diego, 1990)

[27] M. T. Yin and M. L. Cohen, Phys. Rev. B **26**, 3259 (1982)

[28] J. Ihm and M. L. Cohen, Phys. Rev. B **21**, 1527 (1980)

[29] J. Stoer and R. Bulirsch, "Introduction to Numerical Analysis", 2nd ed. (Springer, New York, 1992)

[30] D. J. Chadi and M. L. Cohen, Phys. Rev. B **8**, 5747 (1973)

[31] C. H. Park, B.-H. Cheong, K. H. Lee, and K. J. Chang, Phys. Rev. B **49**, 4485 (1994)

[32] J. Callaway, "Quantum Theory of the Solid State", 2nd. ed., (Academic, San Diego, 1991).

[33] C. -L. Fu and K. -M. Ho, Phys. Rev. B **28**, 5480 (1983)

[34] M. R. Pederson and K. A. Jackson, Phys. Rev. B **43**, 7312 (1991)

[35] M. T. Yin, *Proc. 17th Int. Conf. on the Physics of Semiconductors*, eds. J. D. Chadi and W. A. Harrison (Springer, NY, 1984) p. 927

[36] O. Jepson and O. K. Anderson, Solid State Commun. **9**, 1763 (1971)

[37] C. Herring, Phys. Rev. **52**, 365 (1937)

[38] G. F. Koster, *Solid Stete Physics* **5**, (1957) p. 173. It is sometimes pointed out that there are errors in the printed character tables in the literature, including this Koster's paper. Errors found in the major printed tables are corrected in J. L. Warren, Rev. Mod. Phys. **40**, 38 (1968).

[39] O. V. Kovalev, "Representations of the Crystallographic Space Groups", 2nd ed., translated to English (Gordon adn Breach Scientific Pub., Yverdon, 1993)

[40] C. J. Bradley and A. P. Cracknell, "The Mathematical Theory of Symmetry in Solids", (Clarendon Press, Oxford, 1972)

[41] L. Verlet, Phys. Rev. **159**, 98 (1967)

[42] Technical Report No. 21 "Plotting of wavefunctions on the real space"

[43] D. Frenkel and B. Smit, "Understanding Molecular Simulation", (Academic, San Diego, 1996)

[44] S. E, Koonin and D. C. Meredith, "Computational Physics", (Addison Wesley, Massachusets, 1990)