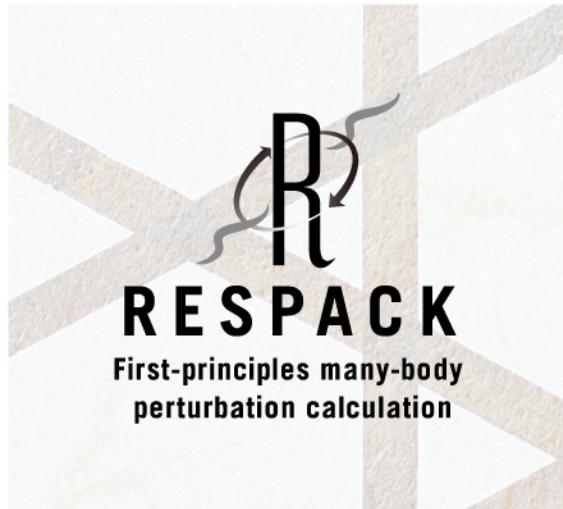


RESPACK Manual

Kazuma Nakamura

(Department of Basic Sciences, Kyushu Institute of Technology)

May 27, 2019



Contents

| | | |
|-----------|--|-----------|
| 1 | Introduction | 4 |
| 1.1 | Program | 4 |
| 1.2 | Unit system | 4 |
| 1.3 | Download source code | 4 |
| 1.4 | Compiling source code | 4 |
| 1.4.1 | Compile using CMake | 4 |
| 1.4.2 | Compile using Makefile | 5 |
| 1.5 | Calculation directory | 5 |
| 2 | Making input files of RESPACK | 6 |
| 2.1 | From xTAPP to RESPACK | 6 |
| 2.2 | From QUANTUM ESPRESSO to RESPACK | 6 |
| 3 | wannier | 8 |
| 3.1 | Input file | 8 |
| 3.1.1 | ¶m_wannier | 8 |
| 3.1.2 | ¶m_interpolation | 10 |
| 3.1.3 | ¶m_visualization | 10 |
| 3.2 | Execute job | 11 |
| 3.3 | Calculation results | 11 |
| 4 | chiqw | 15 |
| 4.1 | Input file (¶m_chiqw) | 15 |
| 4.2 | Execute job | 16 |
| 4.3 | Calculation results | 16 |
| 5 | calc_w3d and calc_j3d | 19 |
| 5.1 | Input file (¶m_calc_int) | 19 |
| 5.2 | Execute job | 19 |
| 5.3 | Calculation results | 19 |
| 6 | Summary: Al calculation input | 23 |
| 7 | Transfer analysis | 24 |
| 7.1 | Usage | 24 |
| 7.2 | Command line | 25 |
| 7.3 | Calculation results | 25 |
| 8 | La₂CuO₄ (Interaction evaluation using constrained RPA) | 28 |
| 9 | TiO₂ (Wannier function calculation with local coordinates) | 31 |
| 10 | Si (Wannier function using B matrix) | 33 |
| 11 | SrVO₃ (Interaction evaluation using constrained RPA) | 35 |
| 12 | (TMTSF)₂PF₆ (Interaction evaluation using constrained RPA)) | 36 |
| 13 | Parallel calculation for q points (chiqw) | 37 |
| 14 | Chiqw calculation for specific q points | 38 |
| 15 | Visualization for specific realspace Wannier orbitals | 39 |

| | |
|--|-----------|
| 16 Pseudopotential dependence of interaction parameters | 40 |
| 17 File format (reading format) | 42 |
| 18 License | 44 |
| 19 How to quote | 44 |

1 Introduction

1.1 Program

RESPACK is a first-principles calculation software for evaluating the interaction parameters of materials, and is able to calculate the maximally localized Wannier functions, the RPA response functions, and frequency-dependent electronic interaction parameters. RESPACK consists of three programs:

1. Wannier function calculation (`wannier`),
2. dielectric function calculation (`chiqw`),
3. electronic interaction calculation (direct integral: `calc_w3d`, exchange integral: `calcj3d`).

These three programs are independent and each program needs

- input file (namelist form)
- files as band-calculation output

RESPACK receives input data from a band calculation using norm-conserving pseudopotentials with plane-wave basis sets. Automatic generation scripts which convert results of the band calculation to inputs for RESPACK are prepared for xTAPP and QUANTUM ESPRESSO codes. RESPACK can be successfully applied to wide materials such as metals, semiconductors, transition-metal compounds, and organic compounds. It supports OPENMP/MPI and can be used in both laboratory-computer environment and System B at Institute for Solid State Physics, the University of Tokyo.

In this manual, we describe procedures to evaluate effective interaction parameters of Al as a benchmark. We also describe procedures of the case of La_2CuO_4 for constrained RPA calculation. Examples of Si, TiO_2 , SrVO_3 , organic compound $(\text{TMTSF})_2\text{PF}_6$ are also presented.

1.2 Unit system

Programs are described in atomic units, but for input and output, Å, Ry, eV, and lattice coordinates are used.

1.3 Download source code

The source code cab be downloaded from the official website <https://sites.google.com/view/kazuma7k6r>. Please download `RESPACK.tar.gz`.

1.4 Compiling source code

Firstly, you unpack the source code `RESPACK.tar.gz` in your machine. In a directory `RESPACK`, you see five directories (i) `sample` containing the inputs of calculation examples, (ii) `src` containing source codes, (iii) `util` containing necessary scripts to supplement calculation, (iv) `config` containing setting files of CMake, and (iv) `GPL` containing the software license.

1.4.1 Compile using CMake

RESPACK can be compiled using CMake. In CMake, user generates a working directory for compilation, and then creates a `Makefile` script using the `cmake` command. Then, the program is compiled using the `make` command. The following is the work flow.

```

tar -zvxf RESPACK.tar.gz
cd RESPACK
mkdir build
cd build
cmake -DCONFIG=gcc -DCMAKE_INSTALL_PREFIX=PATH_TO_INSTALL ../
make
make install

```

If `make` is successful, executable files are created in each directory under `RESPACK/build/src`. By executing `make install`, the executable files will be installed in the `bin` directory under the directory specified with the `-DCMAKE_INSTALL_PREFIX` option. If the `-DCMAKE_INSTALL_PREFIX` option is omitted, it will be installed under `/usr/local`. Also, the `-DCONFIG` option are used for reading CMake configuration files stored in the `config` directory. The following options are available

- sekirei: System B (“sekirei”) in Institute Solid State Physics, The University of Tokyo
- intel: Intel compiler
- gcc: GNU compiler

If you want to change the source-code compile or re-execute the `cmake` command, it is recommended that you delete the working directory itself (`build`) and restart, because the previous settings may remain.

Another point to note is that `Cmake` in an environment that uses the INTEL MPI library. In this case it is better to specify the compiler directly; specify `mpiifort` of INTEL MPI compiler with the `-DMPI_Fortran_COMPILER` option below.

```
cmake -DMPI_Fortran_COMPILER=mpiifort -DCMAKE_INSTALL_PREFIX=PATH_TO_INSTALL ../
```

1.4.2 Compile using Makefile

Instead of using `CMake`, user can compile using `Makefile` which is prepared for each program. When moving to `src`, user finds three source directories; `calc_int`, `chiqw`, `wannier`. `Makefile` for source compilation are prepared in each source directory. User moves to the source directory and executes `make`. In the case of `chiqw`, the work so far is as follows:

```

tar -zvxf RESPACK.tar.gz
cd RESPACK/src/chiqw/
make

```

After `make`, an executable file `calc_chiqw` is generated. This procedure is the same for other programs.

1.5 Calculation directory

In this tutorial, the directory in which the calculation is performed is set to the current directory `./` and described as `calculation_directory`. User moves the executable files generated via `make` in the current directory. After the RESPACK calculations, the calculation results are accommodated in `./dir-xxx`. This is the preparation stage for calculation.

2 Making input files of RESPACK

In RESPACK, band calculations must be executed in advance. In this section, we describe the procedure to connect the band calculation software xTAPP and QUANTUM ESPRESSO to RESPACK. Details of the band calculation are given in the official manual of each software. Here, we focus on the procedures necessary for RESPACK calculation.

2.1 From xTAPP to RESPACK

After xTAPP band calculation, we generate input files of RESPACK calculation with a script. First, we move to directory `RESPACK/util/xtapp2respack` and compile source codes. Executable files `wfn2respack` and `strconv` are generated. These `wfn2respack`, `strconv`, and a script file `xtapp2respack.sh` are necessary for convert. Then, we move back to the directory where band calculation was executed and copy these above three files to the calculation directory. With these files, input files for RESPACK are generated from a wave function file (`@@@.wfn`) and a structure file (`@@@.str`) (where `@@@` specifies the material name in xTAPP). After executing the script, the directory `./dir-wfn` is created and 9 files named `dat.xxx` are generated in this directory. For example, in the case of Al calculation, the work so far is as follows:

```
cd RESPACK/util/xtapp2respack/
make
cp wfn2respack calculation_directory/.
cp strconv calculation_directory/.
cp xtapp2respack.sh calculation_directory/.
cd calculation_directory/
./xtapp2respack.sh -b ./wfn2respack -s ./strconv Al
```

Each file to be generated is as follows. For the format of each file, see the [17](#) chapter.

1. `./dir-wfn/dat.bandcalc` (band calculation information)
 2. `./dir-wfn/dat.sample-k` (sample k points)
 3. `./dir-wfn/dat.nkm` (the number of expanded G vector)
 4. `./dir-wfn/dat.eigenvalue` (energy eigenvalue)
 5. `./dir-wfn/dat.kg` (reciprocal lattice G vector)
 6. `./dir-wfn/dat.wfn` (wave function)
 7. `./dir-wfn/dat.lattice` (lattice vector)
 8. `./dir-wfn/dat.symmetry` (Symmetry information)
 9. `./dir-wfn/dat.atom_position` (Atomic position)
-

The above treatment is the connection procedure from xTAPP band calculation to RESPACK.

2.2 From QUANTUM ESPRESSO to RESPACK

After QUANTUM ESPRESSO band calculation, we generate input files of the RESPACK calculation as follows. This procedure uses Python script `qe2respack.py` in the directory `RESPACK/util/qe2respack`. We first move the calculation directory, and execute the script as follows:

```
python qe2respack.py outdir/prefix.save/
```

Here, `./outdir/prefix.save/` is the directory containing the output of QUANTUM ESPRESSO band calculation. After this procedure, the directory `./dir-wfn` is created under the calculation directory where the python script was run. 9 files named `dat.xxx` (see [2.1](#)) are generated. If `./dir-wfn` already exists, it will be copied to `./dir-wfn.backup/`. At that time, if `./dir-wfn.backup/` exists, it will be deleted.

Also note that the step of performing `qe2respack.py` is always performed immediately after "SCF calculation" which is performed with a variable `calculation='scf'` in namelist `&control` of QUANTUM ESPRESSO. This is because RESPACK needs the data for the irreducible k points. The above treatment is the connection procedure from QUANTUM ESPRESSO band calculation to RESPACK.

3 wannier

Here, we describe the procedure of Wannier function calculation (`wannier`). The calculation conditions are $6 \times 6 \times 6$ k -point sampling (the total number of the irreducible k points is 16) and 36-Ry wave function cutoff. The total number of bands to be calculated is 50 (occupied band number 1, Partial occupied band number 1, unoccupied band number 48, including a band up to about 120 eV above the Fermi level).

3.1 Input file

Here, we explain the input file `input.in` for the Wannier function calculation of Al. The input file uses the FORTRAN namelist format. The input of the Wannier function calculation consists of three namelists `¶m_wannier` (condition of the Wannier function calculation), `¶m_interpolation` (condition of the Wannier-interpolation band calculation), and `¶m_visualization` (condition of visualization of the realspace Wannier functions).

3.1.1 ¶m_wannier

The namelist `¶m_wannier` contains parameters for the Wannier function calculation. Parameters can be arranged freely. Parameters in the namelist are set to default values if not defined. The explanation of each parameter is written in the right of !. A mark “n” in the explanation part indicates that the parameter is a required variable, and an error occurs if you forget to input. A parameter written as “o” represents an optional parameter, and if you omit the input, the default value is set.

```
&param_wannier
N_wannier=9,                                     !n: Number of Wannier functions you want to calculate
Lower_energy_window=-10.0d0,                      !n: Lower limit of energy window (eV)
Upper_energy_window= 35.0d0,                        !n: Upper limit of energy window (eV)
N_initial_guess=9,                                 !n: Number of initial Gaussian
icell=0,                                         !o: Crystal lattice information
EPS_SPILLAGE=1.0d-4,                            !o: Convergence threshold in spillage minimization (au)
EPS_SPREAD=1.0d-6,                             !o: Convergence threshold in spread minimization (au)
DAMP=0.1d0,                                       !o: Convergence factor in spillage minimization
MAX_STEP_LENGTH=4.0d0,                           !o: Step length in spread minimization
set_inner_window=F,                            !o: Whether to set the inner window
Lower_inner_window=10.0d0,                         !o: Lower limit of energy inner window (eV)
Upper_inner_window=35.0d0,                         !o: Upper limit of energy inner window (eV)
flg_initial_guess_direc=0,                       !o: Coordinate system displaying initial Gaussian
flg_BMAT=0,                                       !o: Whether to set linear coefficients of initial Gaussian
electron_number_wannier_space=3.0,!o: Electron number in Wannier space
flg_fermisurface=1,                            !o: Fermisurface calculation (do not: 0, do :1)
flg_global_dos=1/                                !o: Density of state calculation (do not: 0, do:1)
s  0.2d0 0.0d0 0.0d0 0.0d0                      !n: Orbital type, Gaussian width, Gaussian position s@Al
px 0.2d0 0.0d0 0.0d0 0.0d0                      !n: px@Al
py 0.2d0 0.0d0 0.0d0 0.0d0                      !n: py@Al
pz 0.2d0 0.0d0 0.0d0 0.0d0                      !n: pz@Al
dxy 0.2d0 0.0d0 0.0d0 0.0d0                     !n: dxy@Al
dyz 0.2d0 0.0d0 0.0d0 0.0d0                     !n: dyz@Al
dz2 0.2d0 0.0d0 0.0d0 0.0d0                     !n: dz2@Al
dzx 0.2d0 0.0d0 0.0d0 0.0d0                     !n: dzx@Al
dx2 0.2d0 0.0d0 0.0d0 0.0d0                     !n: dx2-y2@Al
```

`¶m_wannier` requires four necessary parameters: `N_wannier`, `Lower_energy_window`, `Upper_energy_window`, `N_initial_guess`. `Lower_energy_window` and `Upper_energy_window` are

variables that specify the energy window. The program makes `N_wannier` pieces of the Wannier functions from this energy range. Figure 1 shows the relationship between band dispersion and energy window. The lower limit of the energy window was set to -10 eV, and the upper limit was set to $+35$ eV. Nine Wannier functions (`N_wannier = 9`) are made from the bands within this window.

`N_initial_guess` is a variable specifying the number of initial Gaussian. We prepare nine in this calculation (`N_initial_guess = 9`). The initial Gaussians are also necessary inputs. This corresponds to "s 0.2d0 0.0d0 0.0d0 0.0d0" to "dx2 0.2d0 0.0d0 0.0d0 0.0d0". We use nine Gaussians of type s, px, py, pz, dxy, dyz, dz2, dxz, dx2-y2 placed on the Al site as initial guess, where x, y, z is the Cartesian coordinates used to represent the Bravais lattice.

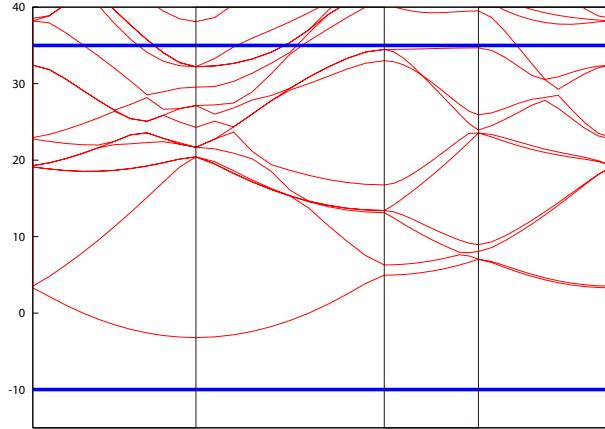


Figure 1: Band structure of Al. The lower limit of the energy window is -10 eV (`Lower_energy_window=-10.0`) and the upper limit is $+35$ eV (`Upper_energy_window=35.0`). From the bands in this window, nine Wannier functions (`N_wannier=9`) were created from nine initial Gaussian (`N_initial_guess=9`).

`icell` is a variable that specifies crystal lattice. `icell=0` is default which means an automatic search. `EPS_SPILLAGE`, `EPS_SPREAD`, `DAMP`, `MAX_STEP_LENGTH` are parameters for convergence of the calculation. These are set to the default values.

`set_inner_window` is a parameter that specifies the inner energy window. `Lower_inner_window` and `Upper_inner_window` are variables that specify the inner energy window. When these parameters are specified, the Wannier functions are constructed so that the bands within the inner energy window coincides with the original bands. In this Al calculation, `set_inner_window=F`, so the inner window is not set. Details of these parameters are described in La_2CuO_4 calculation (chapter 8).

`flg_initial_guess_direc` is a variable specifying the coordinate system for displaying the Gaussian function. If you want to change to local coordinates instead of the Cartesian coordinates, you set `flg_initial_guess_direc=1`. This flag is explained in TiO_2 calculation in the 9 chapter.

`flg_BMAT` is a variable related to the procedure for making the initial guess of the Wannier function. If `flg_BMAT=1`, we linearly combine the Gaussians to make initial guess as follows:

$$w_j^{in}(\mathbf{r}) = \sum_i B_{ij} g_i(\mathbf{r}).$$

If `flg_BMAT=0`, the linear combination is not performed, and hence the B matrix is an identity matrix. If `flg_BMAT=1` is set, the coefficient B_{ij} of the linear combination is read following the information of the initial Gaussian. This will be explained in Si calculation in the 10 chapter.

The variable `electron_number_wannier_space` specifies the number of electrons in the Wannier space. If this variable is set, the Fermi level is recalculated. The Fermi level is estimated so that the number of electrons (3.0 in the present example) matches the numerical integral for the density

of states until the Fermi level. The default is 0.0 and in this case the Fermi level obtained in the band calculation is employed.

Set `flg_global_dos=1` if you want to calculate the density of states for the band energy obtained in the band calculation. The default is 0 and it is not calculated. The density of states in the Wannier space is always calculated.

Set `flg_fermisurface=1` if you want to calculate the Fermi surface. The input of the Fermi-surface drawing software FERMISURFER is output. The default is 0 and it is not calculated.

3.1.2 ¶m_interpolation

The namelist `¶m_interpolation` contains input variables for drawing the Wannier-interpolation band. The interpolation-band calculation is always performed, so variables in this namelist are almost required parameters. An error occurs if you forget to specify the required parameters. Variable explanations are indicated in the right of !.

```
&param_interpolation
N_sym_points=5,    !n: Total number of symmetric k points in calculation lines
Ndiv=40/           !o: Number of divisions between symmetric k points
0.500 0.500 0.500 !n: Symmetric k point; SK_sym_pts(1:3,1); L
0.000 0.000 0.000 !n: Symmetric k point; SK_sym_pts(1:3,2); G
0.500 0.000 0.500 !n: Symmetric k point; SK_sym_pts(1:3,3); X
0.500 0.250 0.750 !n: Symmetric k point; SK_sym_pts(1:3,4); W
0.500 0.500 0.500 !n: Symmetric k point; SK_sym_pts(1:3,5); L
```

In this namelist, you specify the number of symmetric k points in the Brillouin zone, `N_sym_points`. `Ndiv` is the number of divisions between two symmetric k points and is an optional parameter. The default value is 40. Below this namelist, give the coordinates of the symmetric k point (in reciprocal lattice coordinates). It is desirable to compare the original bands obtained by the band calculation with the Wannier-interpolation bands, and to check whether the latter is well reproduced the former.

3.1.3 ¶m_visualization

The namelist `¶m_visualization` contains input variables for drawing the realspace Wannier function. When calculation was completed, the realspace Wannier function data for drawing in visualization software VESTA (<http://jp-minerals.org/vesta/jp/>) is generated.

```
&param_visualization
flg_vis_wannier=1,!o: Calculate realspace Wannier function (do not: 0, do 1)
N_write_wannier=9,!o: Total number of realspace Wannier functions to be calculated
ix_vis_min=-1,      !o: Draw range of Wannier function (starting lattice in a1 direction)
ix_vis_max= 1,      !o: Draw range of Wannier function (end lattice in a1 direction)
iy_vis_min=-1,      !o: Draw range of Wannier function (starting lattice in a2 direction)
iy_vis_max= 1,      !o: Draw range of Wannier function (end lattice in a2 direction)
iz_vis_min=-1,      !o: Draw range of Wannier function (starting lattice in a3 direction)
iz_vis_max= 1,      !o: Draw range of Wannier function (end lattice in a3 direction)
```

If you want to calculate the realspace Wannier function, you set `flg_vis_wannier=1`. The default is 0, which does not perform calculation. `N_write_wannier` is the number of the realspace Wannier functions you want to calculate. By default, this variable is set to `N_wannier`. You can specify fewer than `N_wannier`. In this case, the program calculates `N_write_wannier` species of the Wannier functions (see chapter 15).

3.2 Execute job

The job is executed as follows. The commands in the sh system are as follows.

```
export OMP_NUM_THREADS=16
export MKL_NUM_THREADS=16
./calc_wannier < input.in > LOG.wannier
```

`calc_wannier` is an executable file for calculating the Wannier function. Jobs are executed with `input.in` placed in the current directory and band-calculation files placed in `./dir-wfn`. This job is `openmp` job (using 16 threads). The standard output of the calculation is output to `LOG.wannier`.

3.3 Calculation results

When the job is executed, the directory `dir-wan` is created under the calculation directory. The calculation result is output in this directory. The output file is as follows.

1. `./dir-wan/dat.h_mat_r` (Realspace transfer matrix)
 2. `./dir-wan/dat.iband` (Wannier-interpolation band)
 3. `./dir-wan/dat.umat` (Unitary matrix)
 4. `./dir-wan/dat.wan` (Expansion coefficients of Wannier function)
 5. `./dir-wan/dat.wan-center` (Position of center of Wannier function)
 6. `./dir-wan/dat.ns-nb` (Band information considered in Wannier-function calculation)
 7. `./dir-wan/dat.wan-realspace-001.grd~dat.wan-realspace-009.grd` (Realspace Wannier function)
 8. `./dir-wan/dat.supercell-002x002x002-Al.cif` (Crystal structure)
 9. `./dir-wan/dat.frmsf` (Fermi surface data)
 10. `./dir-wan/dat.dos.global-total` (Density of state for band energies)
 11. `./dir-wan/dat.dos.wan-total` (Density of states in Wannier space)
 12. `./dir-wan/dat.dos.wan-001~dat.dos.wan-009` (Partial density of states)
-

`dat.h_mat_r` contains data for the realspace transfer matrix: the Kohn-Sham Hamiltonian matrix elements with the Wannier function $\langle w_{i0} | h_{KS} | w_{jR} \rangle$. The range of the lattice \mathbf{R} is $-Na_1:Na_1$ along the a_1 direction, $-Na_2:Na_2$ along the direction of a_2 , $-Na_3:Na_3$ along the a_3 direction (The values of Na_1 , Na_2 , Na_3 are automatically set from the sample- k point number).

In this program, the i th Wannier function of the lattice \mathbf{R} is defined as follows:

$$w_{i\mathbf{R}}(\mathbf{r}) = \frac{1}{\sqrt{N}} \sum_{\mathbf{k}}^N e^{-i\mathbf{k}\mathbf{R}} \sum_{\alpha=N_s(\mathbf{k})}^{N_b(\mathbf{k})} U_{\alpha i}^{\mathbf{k}} \psi_{\alpha\mathbf{k}}(\mathbf{r}), \quad (1)$$

where N is the total number of Monkhorst-Pack k mesh. $N_s(\mathbf{k})$ is the starting point number of the bands used to make the Wannier function, and $N_b(\mathbf{k})$ is the total number of those bands, which are automatically calculated from the energy-window information. $N_s(\mathbf{k})$, $N_b(\mathbf{k})$ are stored

in `dat.ns-nb`, and the transformation matrix $\{U_{\alpha i}^k\}$ is contained in `dat.umat`. Note that these quantities depend on the k point. $\psi_{\alpha k}(\mathbf{r})$ is the Bloch wave function and is defined as follows:

$$\psi_{\alpha k}(\mathbf{r}) = \frac{1}{\sqrt{N}} \sum_{\mathbf{G}} c_{\mathbf{k}+\mathbf{G},\alpha} \frac{1}{\sqrt{\Omega}} e^{i(\mathbf{k}+\mathbf{G}) \cdot \mathbf{r}}, \quad (2)$$

where Ω is the volume of the calculation cell. Both the Wannier function $w_{i\mathbf{R}}(\mathbf{r})$ and the Bloch function $\psi_{\alpha k}(\mathbf{r})$ are normalized in the crystal volume. When one inserts Eq. (2) into the Eq. (1), the Wannier function at the home cell ($\mathbf{R}=\mathbf{0}$) is written as

$$w_{i\mathbf{0}}(\mathbf{r}) = \frac{1}{N} \sum_{\mathbf{k}}^N \sum_{\mathbf{G}} \tilde{c}_{\mathbf{k}+\mathbf{G},i} \frac{1}{\sqrt{\Omega}} e^{i(\mathbf{k}+\mathbf{G}) \cdot \mathbf{r}} \quad (3)$$

with

$$\tilde{c}_{\mathbf{k}+\mathbf{G},i} = \sum_{\alpha=N_s(\mathbf{k})}^{N_b(\mathbf{k})} U_{\alpha i}^k c_{\mathbf{k}+\mathbf{G},\alpha}. \quad (4)$$

Here, $\{w_{i\mathbf{0}}(\mathbf{r})\}$ are stored in `dat.wan-realspace-@@@.grd` (@@@ is the number of the output Wannier function, i). Also, $\{\tilde{c}_{i\mathbf{k}+\mathbf{G}}\}$ are stored in `dat.wan`.

`dat.iband` is the Wannier-interpolation band and, in `gnuplot`, is plotted as follows [Fig. 2(a)]. The density of states (`dat.dos.global-total`) and the Wannier density of states (`dat.dos.wan-total`) are also plotted with `gnuplot`. These are shown in Fig. 2(b). The third column of `dat.dos.global-total` and/or `dat.dos.wan-total` contains data of the step function $H(E_F - \omega)$. The unit of the density of states is `states/eV`, which is defined as the total number of states including the spin degeneracy 2.

```
gnuplot> plot 'dir-wan/dat.iband' w line
gnuplot> plot 'dir-wan/dat.dos.global-total', 'dir-wan/dat.dos.global-total' u 1:3,
           'dir-wan/dat.dos.wan-total'
```

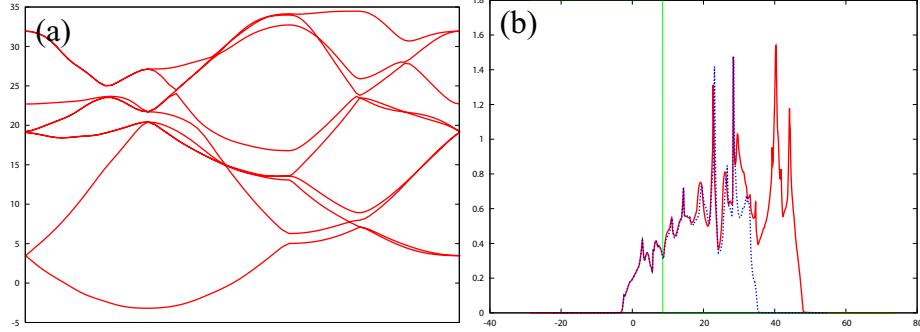


Figure 2: Calculated Wannier interpolation band. Unit of vertical axis is eV. (b) Calculated density of states (red: global, green: step function at Fermi level, blue: the Wannier density of states). The unit of the horizontal and vertical axes are eV and `states/eV`, respectively.

The plot for the partial density of states is also shown in Fig. 3 with `gnuplot`. Only the contributions of the s and $d_{x^2-y^2}$ orbitals are shown.

```
gnuplot> plot 'dir-wan/dat.dos.wan-total', 'dir-wan/dat.dos.wan-001' u 1:3,
           'dir-wan/dat.dos.wan-009'
```

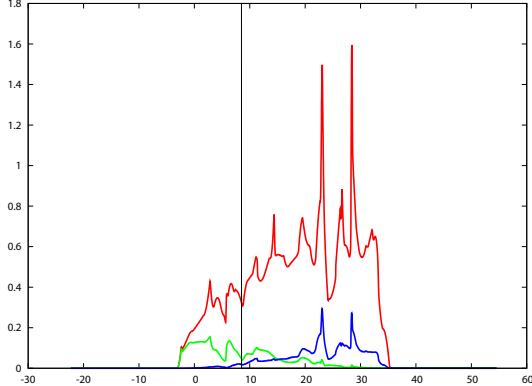


Figure 3: Partial density of states: Red is the Wannier density of state, green is the contribution from the s orbital, and blue is the contribution from the $d_{x^2-y^2}$ orbital. The unit of the horizontal and vertical axes are eV and states/eV, respectively.

Figure 4 displays the atomic structure and the realspace Wannier function with VESTA: `dat.supercell-002x002x002-Al.cif` is a structure file whose size matches the display range of the Wannier function. In this case, the size is $2 \times 2 \times 2$. `dat.wan-realspace-number.grd` is the realspace Wannier function data (*number* is the number of Wannier function). This figure can be created simply by reading `dat.supercell-002x002x002-Al.cif` and `dat.wan-realspace-number.grd` into VESTA.

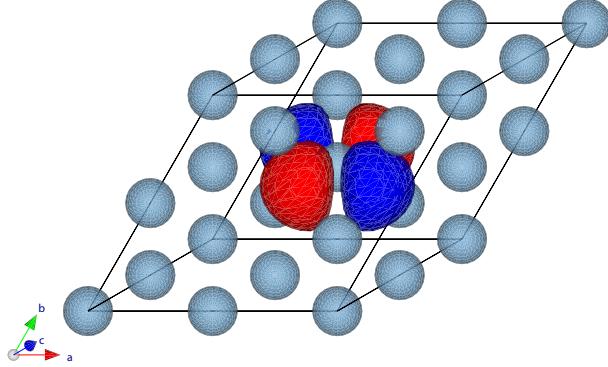


Figure 4: Real space Wannier function. The 9th Wannier function (`dat.wan-realspace-009.grd`) is drawn, which is the $d_{x^2-y^2}$ orbital. Atomic structure data `dat.supercell-002x002x002-Al.cif` and $d_{x^2-y^2}$ Wannier function data (`dat.wan-realspace-009.grd`) are superposed. We use the visualization software VESTA (<https://jp-minerals.org/vesta/jp/>).

Figure 5 displays the calculated Fermi surface with the Fermi surface drawing software FERMISURFER. `dat.frmsf` is the Fermi surface data and is output in the reading form of FERMISURFER. If FERMISURFER is installed, you can make this plot simply by clicking `dat.frmsf`.

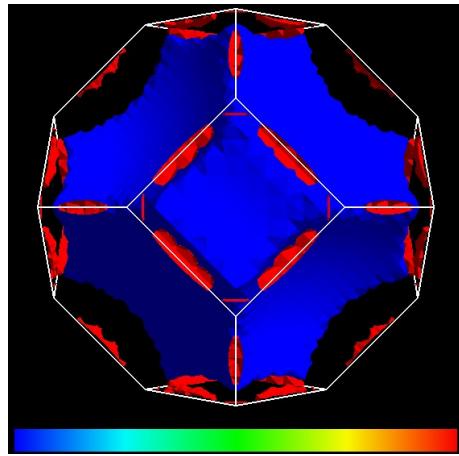


Figure 5: The Fermi surface drawn by the Fermi surface drawing software FERMISURFER (<https://ja.osdn.net/projects/fermisurfer/>). In this calculation, we used dense k -point sampling of $21 \times 21 \times 21$.

4 chiqw

4.1 Input file (¶m_chiqw)

Below is the input file `input.in` for the dielectric function calculation of Al. `¶m_chiqw` is the name of the namelist, and variables are listed below this line. A brief description of variables is given in the right of `!`. In `¶m_chiqw`, all variables are optional parameters, and the user may omit the input. Default values are presented in this list.

```
&param_chiqw
Ecut_for_eps=3.6d0,          !o: Cutoff of polarization function (Ry)
Num_freq_grid=70,            !o: Total number of calculation frequencies
N_CALC_BAND=50,              !o: Total number of bands considered in polarization calc.
MPI_num_proc_per_qcomm=6,    !o: Number of MPI process per community
MPI_num_qcomm=1,              !o: Number of communities
MPI_io_rank=0,                !o: MPI number to be outputted to standard output
shift_ef=0.0d0,               !o: Artificial shift of Fermi level (eV)
Max_excitation_energy=200.0d0,!o: Maximum excitation energy to be considered in calc. (eV)
Green_func_delt=0.1d0,        !o: Smearing value in tetrahedron calculation (eV)
ttrhdrn_dmna=1.0d-3,         !o: Degeneracy judge parameter in tetrahedron calc. (eV)
ttrhdrn_dmnr=1.0d-3,         !o: Degeneracy judge parameter in tetrahedron calc. (eV)
flg_cRPA=0,                  !o: constrained RPA option (0: RPA, 1: cRPA)
flg_calc_type=0              !o: 0: all-q-calc, 1: gamma-only, 2: bulk job
```

`Ecut_for_eps` is the cutoff energy of the polarization function (unit: Ry). By default, it is set to 1/10 of the cutoff energy of the wave function used in the band calculation (in the Al case, since the wave function cutoff is 36 Ry, the polarization function cutoff is set to 3.6 Ry). `Num_freq_grid` is the total number of calculation frequencies of the polarization function. By default, a log grid of 70 points is generated. `N_CALC_BAND` is the number of bands considered in the polarization calculation. By default, the number of bands calculated in the band calculation is set. From this number of bands, the maximum excitation energy E_{max} is determined. The calculated frequency is taken up to $3E_{max}$ by considering the Lorentz tail of the polarization function.

`MPI_num_proc_per_qcomm`, `MPI_num_qcomm`, `MPI_io_rank` are parameters for MPI calculation. In the case of the above setting value, 6 MPI calculation is assumed, and the output is in charge of the master MPI. Parallel computation for q points can be performed by setting the variable `MPI_num_qcomm`, where `q` specifies the irreducible wavevector for the polarization function. This is discussed in more detail in Chapter 13.

`Green_func_delt`, `ttrhdrn_dmna`, `ttrhdrn_dmnr` are parameters for tetrahedron calculation. All default values are prepared. `shift_ef` is used for artificially shift of the Fermi level (unit is eV). `Max_excitation_energy` is set when you want to limit excitation energy (unit is eV).

`flg_cRPA` is a variable that specifies whether or not constrained RPA calculation is performed. By default, normal RPA calculation is performed, so `flg_cRPA=0`. When performing constrained RPA calculation (`flg_cRPA=1`), there are two strategies. One is to specify the exclusion of the polarization by the Wannier functions, and the other is to specify the exclusion of the polarization by specifying the energy window. The default is the former (if you want to use the latter option please contact me). In the case of the former, the `wannier` calculation (Chapter 3) must be executed before the `chiqw` calculation. The constrained RPA calculation will be described in detail in the subsequent La2CuO4 calculation (Chapter 8).

An option to specify the calculation q point is also provided (`flg_calc_type=2`). Details are described in Chapter 14.

4.2 Execute job

The job is executed as follows. Specifications in sh system are as follows.

```
export OMP_NUM_THREADS=16
export MKL_NUM_THREADS=16
mpirun -np 6 ./calc_chiqw < input.in > LOG.chiqw
```

`calc_chiqw` is an executable file for dielectric function calculation. `calc_chiqw` is created by compiling in the directory `RESPACK/src/chiqw` (see the [1.4](#) section). The job is executed with input file `input.in` and files in the directory `./dir-wfn`. This job is a hybrid parallel job [total MPI processes are 6 (`-np 6`) and each MPI process consists of 16 threads]. If you omit `MPI_num_proc_per_qcomm` and `MPI_num_qcomm`, `MPI_num_proc_per_qcomm=6` and `MPI_num_qcomm=1` are automatically set. `input.in` is read as standard input. Standard output of calculation is output to `LOG.chiqw` (standard output is in charge by MPI master by default).

4.3 Calculation results

When the job is executed, a directory `dir-eqs` is created under the calculation directory. In this directory, frequency-grid information (`dat.wgrid`), polarization-function-cutoff information (`dat.chi_cutoff`), electron-energy-loss spectrum (`dat.eels-x, -y, -z`), optical-absorption spectrum (`dat.optical_conductivity-x, -y, -z`), reflectivity (`dat.reflectivity-x, -y, -z`) are output. Here, `-x, -y, -z` attached to the spectrum represents a Cartesian coordinates. Furthermore, the directory `qxxx` attached with the q point number (`xxx`) is generated under the `dir-eqs` directory. Calculation results for each q point are output in this directory. Note that the number of q points (i.e., the number of the `qxxx` directories) is the number of irreducible q points (16 in the Al calculation). For the files in each `qxxx` directory, for example, in the case of the 16th q point, the following output files are generated in the `q016` directory.

1. `./dir-eps/dat.wgrid` (Frequency grid)
 2. `./dir-eps/dat.chi_cutoff` (Polarization cutoff)
 3. `./dir-eps/dat.ttrhdrn` (Tetrahedron parameters)
 4. `./dir-eps/dat.eels-x (-y, -z)` (Electron-energy-loss spectrum)
 5. `./dir-eps/dat.optical_conductivity-x (-y, -z)` (Optical-absorption spectrum)
 6. `./dir-eps/dat.reflectivity-x (-y, -z)` (Reflectance spectrum)
 7. `./dir-eps/q016/dat.log.400` (Calculation end flag)
 8. `./dir-eps/q016/dat.epsqw.600` (Inverse dielectric function data)
 9. `./dir-eps/q016/dat.epsqw.600001~dat.epsqw.600070` (Diagonal terms of inverse dielectric matrix for each frequency)
-

`dat.log.400` is the calculation end flag. Inverse dielectric matrix $\{\epsilon_{\mathbf{G}\mathbf{G}'}^{-1}(\mathbf{q}, \omega)\}$ is accommodated in `dat.epsqw.600` in the binary format. For the files of 600000 series (`dat.epsqw.600000~dat.epsqw.600070`), we store the diagonal term data of the inverse dielectric matrix for each frequency grid. The first column is $|\mathbf{q} + \mathbf{G}|$ (unit is Bohr $^{-1}$), the second column is $\epsilon_{\mathbf{G}\mathbf{G}'}^{-1}\mathbf{q}, \omega_i$, and the third column is the imaginary part. In the case of this job (`Num_freq_grid=70`), it is generated from `dat.epsqw.600001` to `dat.epsqw.600070`. For example, `dat.epsqw.600001` contains the dielectric matrix at the

first frequency ω_1 . Also, `dat.epsqw.600070` contains the diagonal term of the dielectric matrix $\epsilon_{\mathbf{G}\mathbf{G}}^{-1}(\mathbf{q}, \omega_{70})$. On the plot of $\epsilon_{\mathbf{G}\mathbf{G}}^{-1}(\mathbf{q}, \omega)$, perform the following procedure. Here we plot the result of $\omega^{-1} = 0$. The result is shown in Fig. 6.

```
cat dir-eps/q0*/dat.epsqw.600001 > epsq0
gnuplot
gnuplot> plot 'epsq0' u 1:2, 'epsq0' u 1:3
```

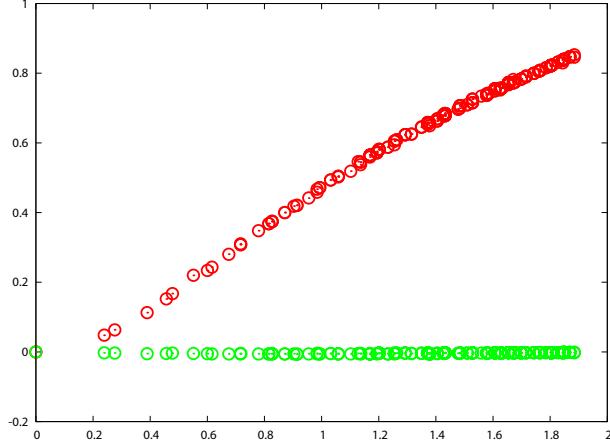


Figure 6: Diagonal terms of the inverse dielectric matrix for all q points $\epsilon_{\mathbf{G}\mathbf{G}}^{-1}(\mathbf{q}, \omega_1)$ (Red is the real part, green is the imaginary part). The horizontal axis is $|\mathbf{q} + \mathbf{G}|$ (Bohr $^{-1}$). In this result, the result at $\omega_1 = 0$ is plotted.

The output of the electronic energy loss spectrum, the optical absorption spectrum, and the reflectance is plotted as follows with `gnuplot`:

```
gnuplot
gnuplot> plot 'dat.eels-x', 'dat.eels-x' u 1:3
gnuplot> plot 'dat.optical_conductivity-x', 'dat.optical_conductivity-x' u 1:3
gnuplot> plot 'dat.reflectivity-x'
```

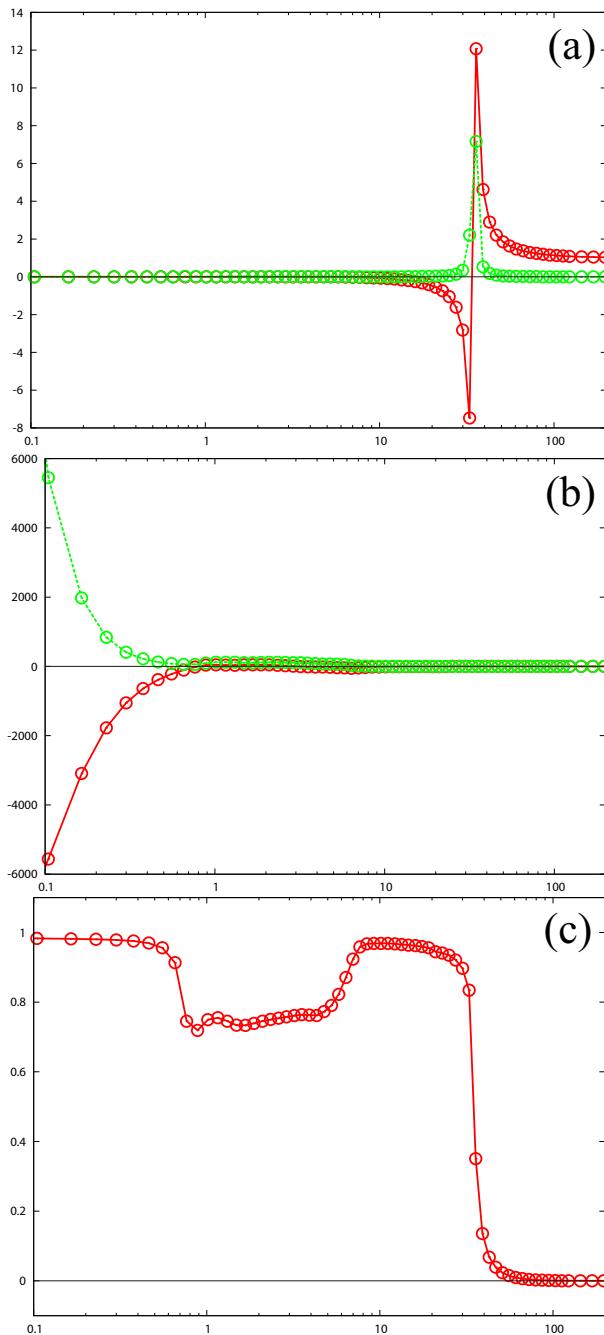


Figure 7: (a) Calculation result for the electronic energy loss spectrum. Red is the real part, green is the imaginary part. The horizontal axis is ω (Hartree). A large peak (green) appears at about 30 eV, which is the intensity due to Al plasmon absorption. (b) Optical absorption spectrum. Red is the real part, green is the imaginary part. (c) Reflectance spectrum.

5 calc_w3d and calc_j3d

Evaluations of direct and exchange integrals are executed in separate executable files (direct-integral calculation is performed with `calc_w3d`, exchange-integral calculation is `calc_j3d`), which can be performed with common inputs. Note that the integral evaluation must be done after the `wannier` and `chiqw` calculation.

5.1 Input file (¶m_calc_int)

Below is the input file `input.in` for the interaction-integral evaluation (`calc_w3d`, `calc_j3d`). Default values are prepared for all variables in the namelist `¶m_calc_int`.

```
&param_calc_int
calc_ifreq=1, !o: Frequency to be calculated
ix_int_min=0, !o: Calc. range of exchange (start lattice in a1 direc.) (calc_j3d only)
ix_int_max=0, !o: Calc. range of exchange (end lattice in a1 direc.) (calc_j3d only)
iy_int_min=0, !o: Calc. range of exchange (start lattice in a2 direc.) (calc_j3d only)
iy_int_max=0, !o: Calc. range of exchange (end lattice in a2 direc.) (calc_j3d only)
iz_int_min=0, !o: Calc. range of exchange (start lattice in a3 direc.) (calc_j3d only)
iz_int_max=0/ !o: Calc. range of exchange (end lattice in a3 direc.) (calc_j3d only)
```

The important point in performing the integral calculation is to calculate the inverse dielectric matrix of all irreducible q points before this calculation. This is because it is impossible to generate all the data on the Monkhorst-Pack mesh if the data at the irreducible q point is insufficient. RESPACK checks the presence/absence of end flag of the dielectric calculation `dat.log.400` in the directories `./dir-eps/qxxx`, and stops calculation when there is not.

5.2 Execute job

The job is executed as follows. Specifications in sh system are as follows.

```
export OMP_NUM_THREADS=16
export MKL_NUM_THREADS=16
./calc_w3d (or ./calc_j3d) < input.in > LOG.W3d (or LOG.J3d)
```

`calc_w3d` or `calc_j3d` are executable files for interaction-integral calculation. The job is executed with the input file `input.in` placed in the current directory and the files placed in the directories `./dir-wfn`, `./dir-eps`, `./dir-wan`. This job is openmp job (using 16 threads). The standard output of `calc_w3d` is output to `LOG.W3d`, and the standard output of `calc_j3d` is output to `LOG.J3d`.

5.3 Calculation results

When the job is executed, `dir-intW` and `dir-intJ` are created under the calculation directory, and the calculation result is output in these directories; the results of `calc_w3d` is contained in `dir-intW`, and the result of `calc_j3d` is contained in `dir-intJ`. The output files are as follows.

-
1. `./dir-intW/dat.Wmat` [Screened direct integral, Eq (5)]
 2. `./dir-intW/dat.Vmat` [Bare direct integral, Eq (7)]
 3. `./dir-intW/dat.VvsR.001 - dat.VvsR.009` ($V(r)$)
 4. `./dir-intW/dat.WvsR.001 - dat.WvsR.009` ($W(r)$)

-
5. `./dir-intW/dat.UvsE.001-001 - dat.UvsE.009-009` ($U(\omega)$)
 6. `./dir-intJ/dat.Jmat` [Screened exchange integral, Eq (6)]
 7. `./dir-intJ/dat.Xmat` [Bare exchange integral, Eq (8)]
 8. `./dir-intJ/dat.JvsE.001-001 - dat.JvsE.009-009` ($J(\omega)$)
-

The screened direct and exchange integrals are defined as follows:

$$W_{ij}(\mathbf{R}, \omega) = \int_V d\mathbf{r} \int_V d\mathbf{r}' w_{i\mathbf{0}}^*(\mathbf{r}) w_{i\mathbf{0}}(\mathbf{r}) W(\mathbf{r}, \mathbf{r}', \omega) w_{j\mathbf{R}}^*(\mathbf{r}') w_{j\mathbf{R}}(\mathbf{r}'), \quad (5)$$

$$J_{ij}(\mathbf{R}, \omega) = \int_V d\mathbf{r} \int_V d\mathbf{r}' w_{i\mathbf{0}}^*(\mathbf{r}) w_{j\mathbf{R}}(\mathbf{r}) W(\mathbf{r}, \mathbf{r}', \omega) w_{j\mathbf{R}}^*(\mathbf{r}') w_{i\mathbf{0}}(\mathbf{r}'). \quad (6)$$

The integral is performed over the crystal volume V . These integrals are matrix element of the screened Coulomb interaction $W(\mathbf{r}, \mathbf{r}', \omega)$ between the i th Wannier function of the home cell ($\mathbf{R}=\mathbf{0}$) and the j th Wannier function of the cell \mathbf{R} . In the case of a bare Coulomb interaction $v(\mathbf{r}, \mathbf{r}') = \frac{1}{|\mathbf{r}-\mathbf{r}'|}$ instead of $W(\mathbf{r}, \mathbf{r}', \omega)$, the bare direct and exchange integrals are defines as follows:

$$V_{ij}(\mathbf{R}) = \int_V d\mathbf{r} \int_V d\mathbf{r}' w_{i\mathbf{0}}^*(\mathbf{r}) w_{i\mathbf{0}}(\mathbf{r}) v(\mathbf{r}, \mathbf{r}') w_{j\mathbf{R}}^*(\mathbf{r}') w_{j\mathbf{R}}(\mathbf{r}'), \quad (7)$$

$$X_{ij}(\mathbf{R}) = \int_V d\mathbf{r} \int_V d\mathbf{r}' w_{i\mathbf{0}}^*(\mathbf{r}) w_{j\mathbf{R}}(\mathbf{r}) v(\mathbf{r}, \mathbf{r}') w_{j\mathbf{R}}^*(\mathbf{r}') w_{i\mathbf{0}}(\mathbf{r}'). \quad (8)$$

Calculated data of `dat.Wmat`, `dat.Jmat`, `dat.Vmat`, `dat.Xmat` are output as a matrix of `N_wannier` \times `N_wannier` of each cell \mathbf{R} . The direct integral is calculated for all \mathbf{R} , while the exchange integral is calculated only in the lattice `ix_intJ_min~iz_intJ_max`, because they are heavy calculations.

`dat.WvsR.number` contains the distance dependence of the static screened direct integral [$W_{ij}(\mathbf{R}, 0)$] (`number` specifies the number of the Wannier function). The direct integrals between the i th Wannier function of the home cell $\mathbf{R}=\mathbf{0}$ and all the Wannier functions at \mathbf{R} are calculated. By default, $\omega = 0$ is calculated (`calc_ifreq=1`), but by changing this variable you can calculate finite ω results. In the Al calculation, `N_wannier=9`, so nine files are output. To see the screened effect, you might plot the bare interaction results `dat.VvsR.number` together. The output is as follows (Fig. 8):

```
gnuplot> plot 'dir-intW/dat.VvsR.001', 'dir-intW/dat.WvsR.001'
```

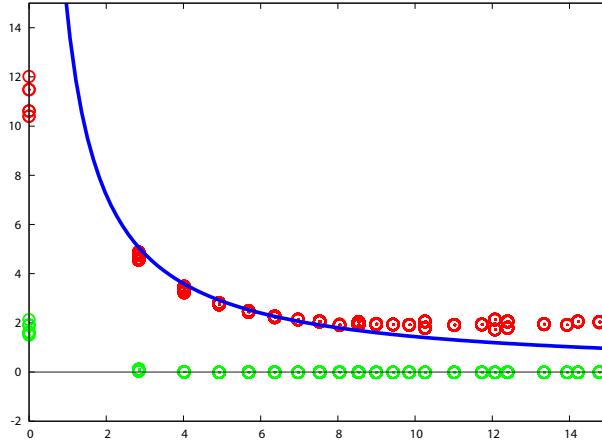


Figure 8: Distance dependence of bare direct integral (red) and static screened direct integral (green). The horizontal axis is the distance (unit: Å) between the Wannier functions $w_{i\mathbf{0}}$ and $w_{j\mathbf{R}}$, and the unit of the vertical axis is eV. This result is for $i=001$. The bare direct integrals are attenuated with $\frac{1}{R}$ (blue), while the static screened direct integrals have values only onsite due to metallic screening.

The frequency dependence of onsite screened direct integrals $W_{ii}(\mathbf{0}, \omega)$ (`dat.UvsE.number-number`) is plotted as follows (Fig. 9). This output is $W_{11}(\mathbf{0}, \omega)$, but, in the program, all $W_{ij}(\mathbf{0}, \omega)$ are calculated for all pairs of the Wannier functions i and j .

```
gnuplot> plot 'dir-intW/dat.UvsE.001-001' u 1:3, 'dir-intW/dat.UvsE.001-001' u 1:4
```

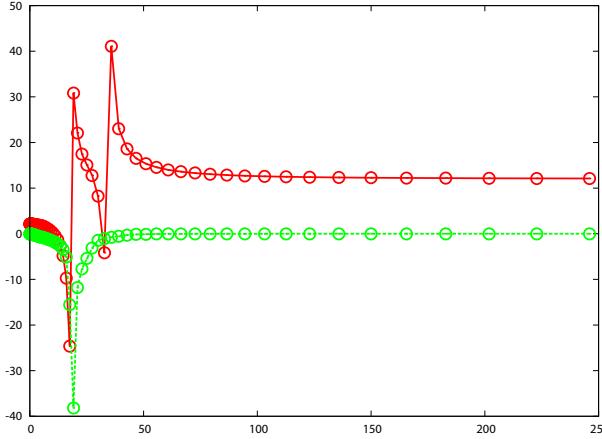


Figure 9: Frequency dependence of onsite screened direct integrals. Red is the real part, and green is the imaginary part. The horizontal axis shows the frequency ω (eV), and the unit of the vertical axis is eV. A pole due to the plasmon excitation can be seen around 20 eV.

The frequency dependence of onsite screened exchange integrals [$J_{ij}(\mathbf{0}, \omega)$] is plotted as follows (Fig. 10). This output is $J_{12}(\mathbf{0}, \omega)$, but, in the program, all $J_{ij}(\mathbf{0}, \omega)$ are calculated for all pairs of the Wannier functions i and j .

```
gnuplot> plot 'dir-intJ/dat.JvsE.001-002' u 1:3, 'dir-intJ/dat.JvsE.001-002' u 1:4
```

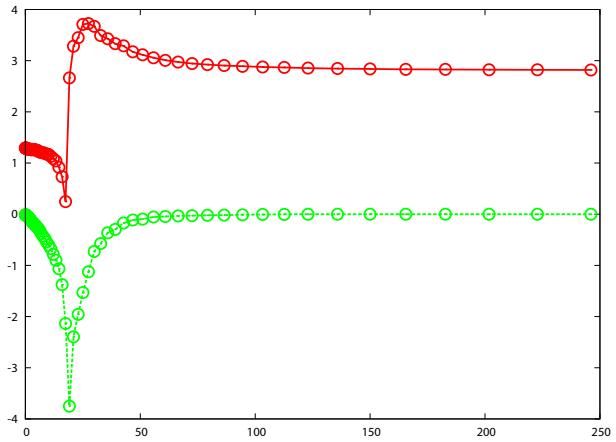


Figure 10: Frequency dependence of onsite screened exchange integrals. Red is the real part, and green is the imaginary part. The horizontal axis shows the frequency ω (eV), and the unit of the vertical axis is eV. We also see a pole around 20 eV, but the overall energy scale is about 1/10 of direct integral.

6 Summary: Al calculation input

We shows the final input of RESPACK calculation for Al, where all the optional variables are omitted. We can easily understand that necessary parameters are only for `wannier` calculation, and other programs are default setting.

```
&param_wannier
N_wannier=9,                      !n: Number of Wannier functions you want to calculate
Lower_energy_window=-10.0d0,!n: Lower limit of energy window (eV)
Upper_energy_window= 35.0d0,!n: Upper limit of energy window (eV)
N_initial_guess=9/                 !n: Number of initial Gaussian
s   0.2d0 0.0d0 0.0d0 0.0d0 !n: Orbital type, Gaussian width, Gaussian position s@Al
px  0.2d0 0.0d0 0.0d0 0.0d0 !n: px@Al
py  0.2d0 0.0d0 0.0d0 0.0d0 !n: py@Al
pz  0.2d0 0.0d0 0.0d0 0.0d0 !n: pz@Al
dxy 0.2d0 0.0d0 0.0d0 0.0d0 !n: dxy@Al
dyz 0.2d0 0.0d0 0.0d0 0.0d0 !n: dyz@Al
dz2 0.2d0 0.0d0 0.0d0 0.0d0 !n: dz2@Al
dzx 0.2d0 0.0d0 0.0d0 0.0d0 !n: dzx@Al
dx2 0.2d0 0.0d0 0.0d0 0.0d0 !n: dx2-y2@Al
&param_interpolation
N_sym_points=5/      !n: Total number of symmetric k points in calculation lines
0.500 0.500 0.500 !n: Symmetric k point; SK_sym_pts(1:3,1); L
0.000 0.000 0.000 !n: Symmetric k point; SK_sym_pts(1:3,2); G
0.500 0.000 0.500 !n: Symmetric k point; SK_sym_pts(1:3,3); X
0.500 0.250 0.750 !n: Symmetric k point; SK_sym_pts(1:3,4); W
0.500 0.500 0.500 !n: Symmetric k point; SK_sym_pts(1:3,5); L
&param_visualization
/
&param_chiqw
/
&param_calc_int
/
```

In this demonstration, the calculation condition is loosened to shorten the calculation time. For convergence, it is necessary to make strict conditions. The Wannier function mainly depends on the density of the k points and the wave function cutoff. The dielectric function depends on the number of unoccupied bands and the dielectric function cutoff. For the Al calculation, it is better to take $10 \times 10 \times 10-k$ points, wave function cutoff 64 Ry, dielectric function cutoff 10 Ry, and unoccupied band number 50. In the case of constrained RPA, it is generally better to include the unoccupied bands up to several tens of eV above the Fermi level.

7 Transfer analysis

When the RESPACK job (`wannier`, `chiqw`, `calc_w3d`, `calc_j3d`) is completed, a directory `./dir-model` is created under the calculation directory, in which the following files are generated:

-
1. `./dir-model/zvo_hr.dat` (transfer integrals)
 2. `./dir-model/zvo_dr.dat` (density matrix)
 3. `./dir-model/zvo_ur.dat` (direct Coulomb integrals)
 4. `./dir-model/zvo_jr.dat` (exchange integrals)
 5. `./dir-model/zvo_geom.dat` (Wannier centers: text format)
 6. `./dir-model/zvo_geom.xsf` (Wannier centers: xsf format)
 7. `./dir-model/zvo_mkkpts.dat` (k-grid information)
 8. `./dir-model/zvo_bandkpts.dat` (k-grid information for band dispersion)
 9. `./dir-model/zvo_ef.dat` (Fermi energy)
-

These files are used for model-parameter analysis. On top of that, these files are inputs for model-solver software (mVMC and $\mathcal{H}\Phi$). For the connection to mVMC and $\mathcal{H}\Phi$, see the site <https://issp-center-dev.github.io/mVMC/docs/index.html> for mVMC and <http://issp-center-dev.github.io/HPhi/index.html> for $\mathcal{H}\Phi$.

RESPACK provides a function to analyze transfer data. Using PYTHON script `tr.py` and the executable files `calc_tr` from FORTRAN code. User can execute the transfer analysis by coping these files to the calculation directory. Preparation for the calculations is as follows:

```
cd RESPACK/src/transfer_analysis
make
cp calc_tr calculation_directory/.
cd RESPACK/util/transfer_analysis
cp tr.py calculation_directory/.
```

7.1 Usage

This program is basically a sorting code for the transfer data. Besides the sorting function, the band-dispersion calculation, the density of states calculation, and the Fermi-surface calculation, etc. can be executed. Shown below are command-line arguments, which can be used in combination:

- Data sort: If the absolute difference in transfers is 0.1 eV or less, sort as the same group.
`tr.py --diff 0.1 (output: dat.tr)`
- Band-dispersion calculation: Cut transfer of 0.1 eV or less.
`tr.py --bnd --cute 0.1 (output: dat.tr, dat.iband)`
- Band-dispersion calculation: Cut transfer of 5 Å or more.
`tr.py --bnd --cutr 5.0 (output: dat.tr, dat.iband)`
- Band-dispersion calculation: Set the k -point grid to $8 \times 8 \times 1$.
`tr.py --bnd --kgrd='8 8 1' (output: dat.tr, dat.iband)`

- Density of states calculation: Set broadening to 0.3 eV.
`tr.py --dos --delt 0.3 (output : dat.tr, dat.dos-total, dat.dos-number)`
- Density of states calculation: Fermi level is calculated with electron number in unitcell as 3.
`tr.py --dos --elec 3.0 (output: dat.tr, dat.dos-total, dat.dos-number)`
- Fermi surface calculation: Set the k -point grid to $40 \times 40 \times 40$.
`tr.py --frm --kgrd='40 40 40' (output: dat.tr, dat.frmsf, dat.orb-number.frmsf)`

7.2 Command line

- **`-h`**
Display argument list.
- **`--bnd`**
The Wannier-interpolated band calculation: The calculation result is output to `dat.iband`.
- **`--dos`**
The density of state calculation: The calculation result is output to `dat.dos-total`. The partial density of states is output to `dat.dos-number`.
- **`--frm`**
The Fermi-surface data generation for FERMISURFER: The result is output to `dat.frmsf`. Orbital contribution of the Fermi surface is output to `dat.orb-number.frmsf`.
- **`--his`**
The histogram calculation for `trans.def`: The calculation result is output to `dat.hist`.
- **`--diff D`**
Set transfer threshold to D (real number) eV: Default is 0.01 eV.
- **`--ecut E`**
Cut transfers below E (real number) eV: Default is 0.00 eV.
- **`--rcut R`**
Cut transfers over R (real number) Å: Default is 100 Å.
- **`--elec N`**
Set the number of electrons in unit cell to N (Real number): Default is no setting.
- **`--delt δ`**
Set broadening for density of state calculations to δ (real number) eV: Default 0.01 eV.
- **`--kgrd=' $n m l$ '`**
Set the k -point grid to $n \times m \times l$ (n , m , and l are integers): Default is the same as the band calculation.

7.3 Calculation results

- data sort
The following is a part of the result of sorting the transfer integral of Al with the following command.

```

python tr.py --ecut 1.0 --diff 0.01      1  5  1  0  0  -1.16784  2.84380
cat dat.tr                                5  1  1  0  0  -1.16784  2.84380
#Threshold for transfer (eV):   1.00
#Threshold for distance (AA): 100.00          irreducible transfer:
#difference in transfers (eV): 0.01           2  2 -1  0  0  1.03877  2.84380
#i,j,ia1,ia2,ia3,tr,dist:                  3  3 -1  0  0  1.03877  2.84380
                                         irreducible transfer:            3  3 -1  0  1  1.03681  2.84380
1  5 -1  0  0  -1.16784  2.84380        4  4 -1  0  1  1.03814  2.84380
                                         2  2 -1  1  0  1.03681  2.84380
5  1 -1  0  0  -1.16784  2.84380        4  4 -1  1  0  1.03814  2.84380
1  6 -1  0  1  1.17119  2.84380        3  3  0 -1  0  1.03681  2.84380
6  1 -1  0  1  1.17119  2.84380        4  4  0 -1  0  1.03814  2.84380
1  8 -1  1  0  1.17119  2.84380        2  2  0 -1  1  1.03877  2.84380
8  1 -1  1  0  1.17119  2.84380        3  3  0 -1  1  1.03877  2.84380
1  6  0 -1  0  -1.17119 2.84380        2  2  0  0 -1  1.03681  2.84380
6  1  0 -1  0  -1.17119 2.84380        4  4  0  0 -1  1.03814  2.84380
1  5  0 -1  1  1.16784  2.84380        2  2  0  0  1  1.03681  2.84380
5  1  0 -1  1  1.16784  2.84380        4  4  0  0  1  1.03814  2.84380
1  8  0  0 -1  -1.17119 2.84380        2  2  0  1 -1  1.03877  2.84380
8  1  0  0 -1  -1.17119 2.84380        3  3  0  1 -1  1.03877  2.84380
1  8  0  0  1  -1.17119 2.84380        3  3  0  1  0  1.03681  2.84380
8  1  0  0  1  -1.17119 2.84380        4  4  0  1  0  1.03814  2.84380
1  5  0  1 -1  1.16784  2.84380        2  2  1 -1  0  1.03681  2.84380
5  1  0  1 -1  1.16784  2.84380        4  4  1 -1  0  1.03814  2.84380
1  6  0  1  0  -1.17119 2.84380        3  3  1  0 -1  1.03681  2.84380
6  1  0  1  0  -1.17119 2.84380        4  4  1  0 -1  1.03814  2.84380
1  8  1 -1  0  1.17119  2.84380        2  2  1  0  0  1.03877  2.84380
8  1  1 -1  0  1.17119  2.84380        3  3  1  0  0  1.03877  2.84380
1  6  1  0 -1  1.17119  2.84380        ...
6  1  1  0 -1  1.17119  2.84380

```

- Energy cutoff dependence for transfer integral

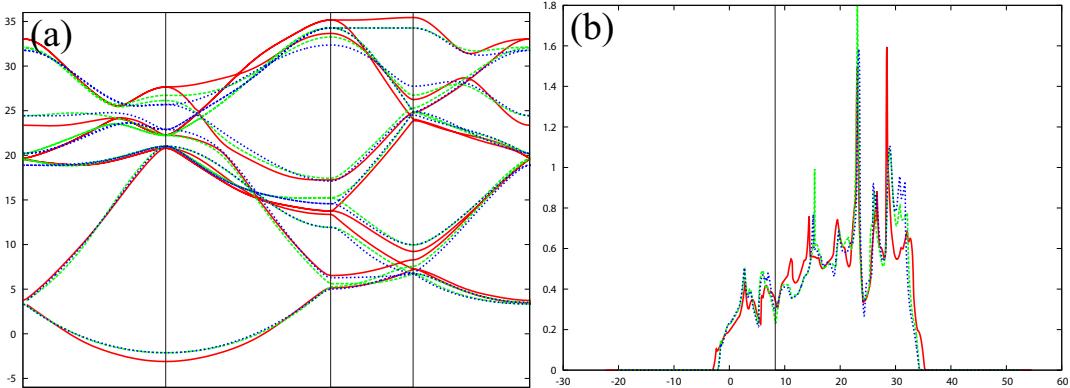


Figure 11: Energy cutoff dependence on transfer integral: (a) Wannier-interpolated band: python tr.py --bnd --ecut=0.0 (red), 0.1 (green), 0.2 (blue). (b) Density of states: python tr.py --dos --ecut=0.0 (red), 0.1 (green), 0.2 (blue).

- K-point grid dependence of the Fermi surface

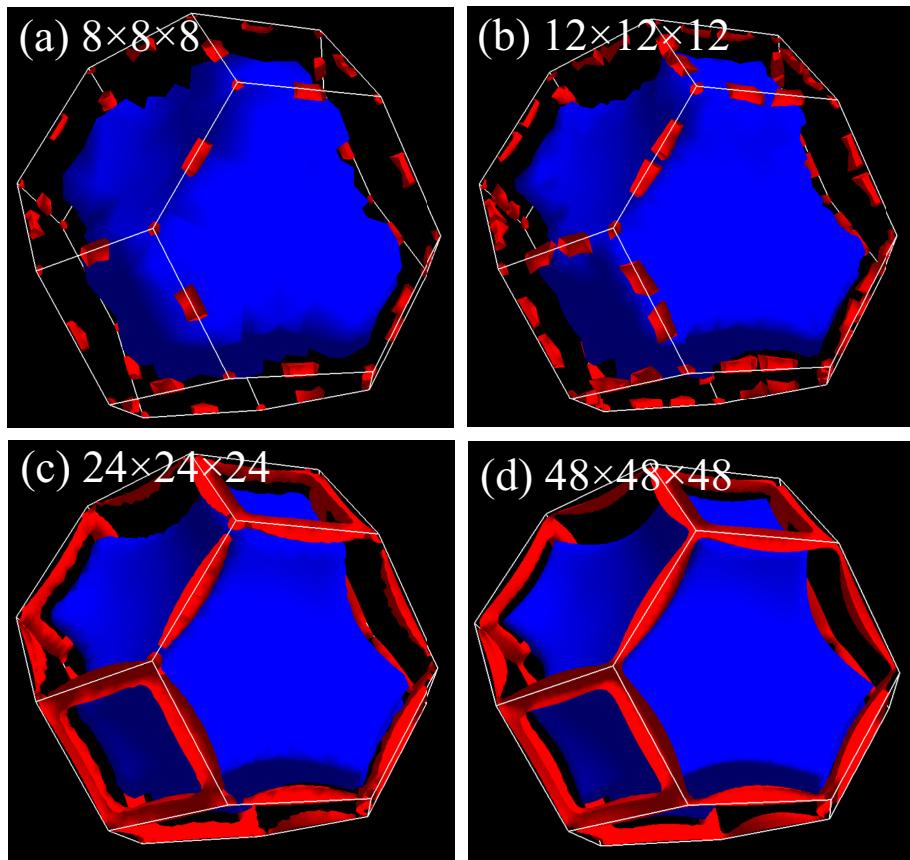


Figure 12: The k -grid dependence of the Fermi surface: `python tr.py --frm --kgrd='8 8 8'` (a), '`12 12 12`' (b), '`24 24 24`' (c), '`48 48 48`' (d).

8 La₂CuO₄ (Interaction evaluation using constrained RPA)

In this chapter, the evaluation of the interaction parameters of the transition-metal oxide La₂CuO₄ is described. In particular, we will describe the setting of the inner energy window in `wannier` calculation and the constrained RPA option in `chiqw` calculation. A condition for the band calculation (using xTAPP) is a $6 \times 6 \times 6$ k -point sampling (30 irreducible k points), a wave function cutoff of 81 Ry, the total number of bands of 60 (26 occupied bands, 1 partial occupied band, 33 unoccupied bands, including the bands up to about 15 eV above the Fermi level).

In this calculation, since the constrained RPA calculation is performed, the variable `flg_cRPA` is set to 1. To reduce the computational cost, the polarization function cutoff is set to 3 Ry (8.1 Ry by default). In case of the constrained RPA, we need to perform the `wannier` calculation before the `chiqw` calculation, because exclusion polarization is defined with the Wannier function. Therefore, the order of calculation is `wannier`→`chiqw`→`calc_w3d` (`calc_j3d`). Below is `input.in` of La₂CuO₄ (body centered tetragonal lattice). In this input, the optional parameters are omitted.

```

&param_wannier
N_wannier=1,                      !Number of Wannier functions you want to calculate
Lower_energy_window=10.0d0,          !Lower limit of energy window (eV)
Upper_energy_window=15.6d0,          !Upper limit of energy window (eV)
N_initial_guess=1,                  !Number of initial Gaussian
set_inner_window=T,                !Setting of inner energy window
Lower_inner_window=12.60d0,          !Energy inner window lower limit (eV)
Upper_inner_window=13.40d0/          !Energy inner window lower limit (eV)
dx2 0.2d0 0.0d0 0.0d0 0.0d0 !Orbital type, Gaussian width, Gaussian position dx2-y2@Cu
&param_interpolation
N_sym_points=6/      !Total number of symmetric k points in calculation lines
0.000 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); G
0.000 0.000 0.500 !Symmetric k point; SK_sym_pts(1:3,1); X
0.250 0.250 0.250 !Symmetric k point; SK_sym_pts(1:3,1); P
0.000 0.500 0.000 !Symmetric k point; SK_sym_pts(1:3,1); N
0.000 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); G
0.500 0.500 -0.500 !Symmetric k point; SK_sym_pts(1:3,1); Z
&param_visualization
flg_vis_wannier=1/ !Calculate realspace Wannier function (Do not: 0, do 1)
&param_chiqw
Ecut_for_eps=3.0d0, !Cutoff of polarization function (Ry)
flg_cRPA=1/         !Constrained RPA option (0: RPA, 1: cRPA)
&param_calc_int
/

```

In the `wannier` calculation, we set the energy window to `Lower_energy_ window=10.0` eV and `Upper_energy_ window=15.6` eV. Note that the values of the energy window depend on the band-calculation software. We make the Wannier function with initial guess of $d_{x^2-y^2}$ centered at the position of the copper atom (`dx2 0.2d0 0.0d0 0.0d0 0.0d0`). In this calculation, the inner window is set (`set_inner_ window=T`). The inner window is set to `Lower_inner_ window=12.6` eV and `Upper_inner_window=13.4` eV. The bands within the inner window is basically reproduced. The inner window is usually set to sandwich the Fermi level (12.97 eV in this calculation).

The resulting Wannier-interpolated band and realspace Wannier function are shown in Fig. 13 and 14, respectively. We show the wavenumber dependence of the static inverse dielectric function by constrained RPA and normal RPA for Fig. 15 (a) and (b), respectively. We also show the electronic energy loss spectrum (eels) by constrained RPA and normal RPA on Figs 16 (a) and (b), respectively. The distance dependence of the statically screened direct integrals is shown in

Fig. 17, where red, green, and blue are bare, constrained-RPA, and normal-RPA direct integrals, respectively. We show the frequency dependence of onsite screened direct integral by constrained RPA and normal RPA for Figs 18 (a) and (b), respectively.

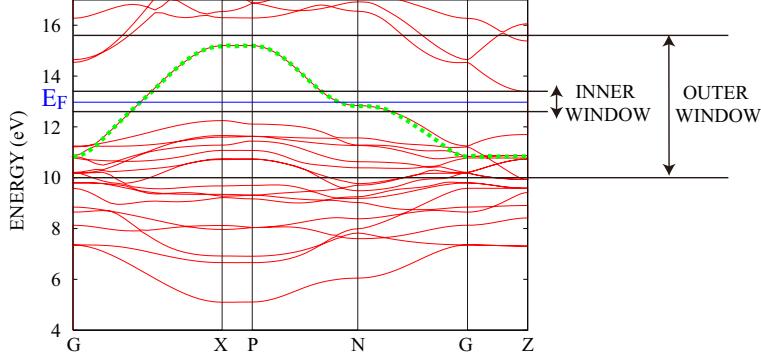


Figure 13: Calculated Wannier interpolated band. The unit of the vertical axis is eV. The lower limit of the energy window was set to 10 eV (`Lower_energy_window=10.0`) and the upper limit was set to 15.6 eV (`Upper_energy_window=15.6`). A single Wannier orbital (`N_wannier=1`) from the bands in this window was made from one initial Gaussian (`N_initial_guess=1`). The inner energy window was set to `Lower_inner_window=12.6` eV and `Upper_inner_window=13.4` eV.

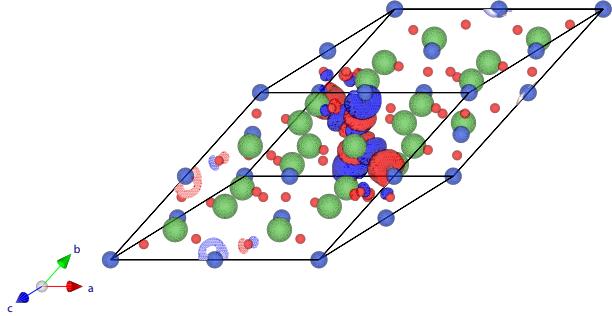


Figure 14: Realspace Wannier function: The Wannier function data is superposed on the structure data whose size matches the display range of the Wannier function ($2 \times 2 \times 2$ supercell).

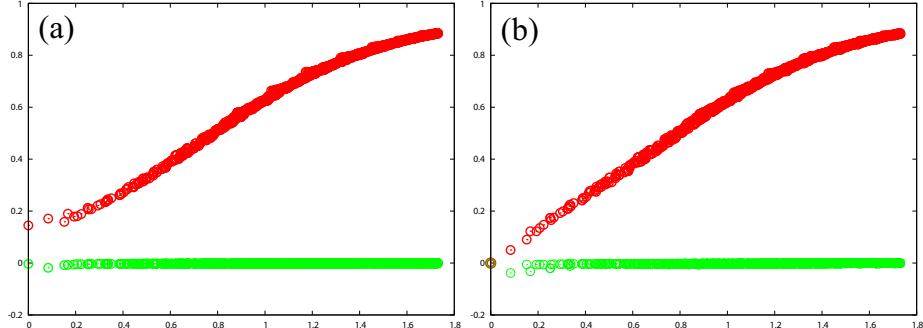


Figure 15: Diagonal data of calculated inverse dielectric matrix with constrained RPA, $\epsilon_{\mathbf{G}\mathbf{G}}^{-1}(\mathbf{q}, \omega_1)$. (Red is the real part, green is the imaginary part). The horizontal axis is $|\mathbf{q} + \mathbf{G}|$ (bohr $^{-1}$). The result of $\omega_1 = 0$ is plotted.

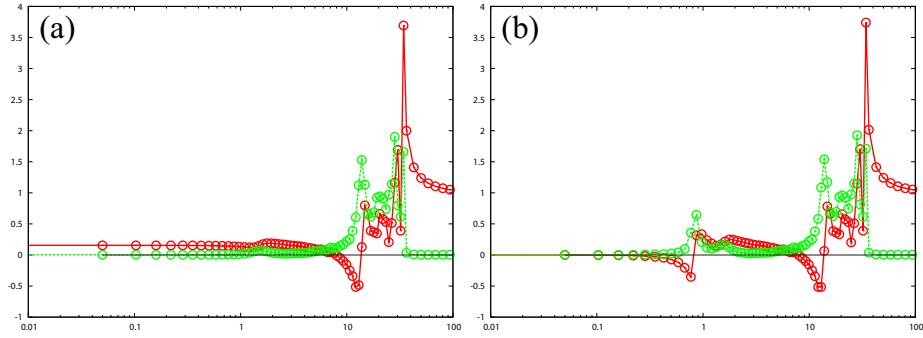


Figure 16: Calculated eels with constrained RPA [$\epsilon_{00}^{-1}(\Gamma, \omega)$] (red is the real part, green is the imaginary part). The horizontal axis is the logarithm of the frequency ω (Hartree).

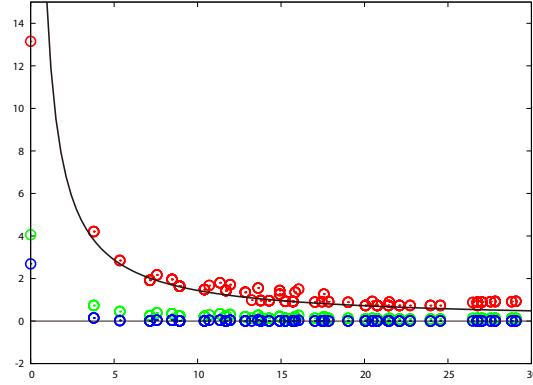


Figure 17: Distance dependence of direct integral [bare (red), constrained-RPA (green), normal-RPA (blue)]. The horizontal axis is the distance (unit: Å) between the Wannier functions w_{i0} and w_{jR} . The unit of the vertical axis is eV. Bare direct integral is attenuated with $\frac{1}{R}$. The constrained-RPA direct integral has finite offsite interactions, while the normal-RPA direct integral has the onsite interaction only because of metal screening.

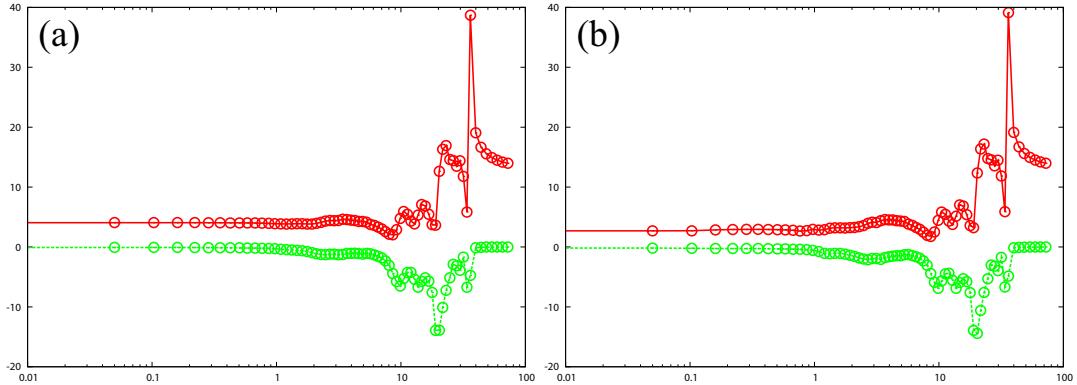


Figure 18: Frequency dependence of onsite screened direct integral with constrained RPA. Red is the real part, and green is the imaginary part. The horizontal axis is the logarithm of the frequency ω (eV). The unit of the vertical axis is eV.

9 TiO₂ (Wannier function calculation with local coordinates)

We next describe the calculation to construct the Wannier function by setting the initial Gaussian along the local coordinates. As an example, we consider constructing the t_{2g} Wannier function of TiO₂ (rutile structure). The following is `input.in` for this calculation. The band calculation is preformed using xTAPP with k -point sampling of $6\times 6\times 6$ (64 irreducible k points) and 64-Ry wavefunction cutoff. Note that the values of the energy window in `¶m_wannier` depend on the band-calculation software.

```

&param_wannier
N_wannier=6,                      !Number of Wannier functions you want to calculate
LOWER_energy_window= 7.0d0,          !Lower limit of energy window (eV)
UPPER_energy_window=11.0d0,          !Upper limit of energy window (eV)
flg_initial_guess_direc=1,          !Display coordinate of initial Gaussian
N_initial_guess=6/                  !Number of initial Gaussian
dxy 0.20 0.00 0.00 0.00  0.46 -0.46 0.76 0.46 -0.46 -0.76  0.71 0.71 0.00 !dxy@Ti1
dyz 0.20 0.00 0.00 0.00  0.46 -0.46 0.76 0.46 -0.46 -0.76  0.71 0.71 0.00 !dyz@Ti1
dzx 0.20 0.00 0.00 0.00  0.46 -0.46 0.76 0.46 -0.46 -0.76  0.71 0.71 0.00 !dzx@Ti1
dxy 0.20 0.50 0.50 0.50 -0.46 -0.46 0.76 0.46  0.46  0.76 -0.71 0.71 0.00 !dxy@Ti2
dyz 0.20 0.50 0.50 0.50 -0.46 -0.46 0.76 0.46  0.46  0.76 -0.71 0.71 0.00 !dyz@Ti2
dzx 0.20 0.50 0.50 0.50 -0.46 -0.46 0.76 0.46  0.46  0.76 -0.71 0.71 0.00 !dzx@Ti2
&param_interpolation
N_sym_points=5/ !Total number of symmetric k points in calculation lines
0.50 0.00 0.50 !Symmetric k point; SK_sym_pts(1:3,1); M
0.00 0.00 0.50 !Symmetric k point; SK_sym_pts(1:3,1); Z
0.00 0.00 0.00 !Symmetric k point; SK_sym_pts(1:3,1); G
0.50 0.00 0.00 !Symmetric k point; SK_sym_pts(1:3,1); X
0.00 0.50 0.00 !Symmetric k point; SK_sym_pts(1:3,1); Y
&param_visualization
FLG_VIS_WANNIER=1/ !Calculate realspace Wannier function (Do not: 0, do 1)

```

In this `wannier` calculation, we set the energy window to `Lower_energy_window=7.0 eV` and `Upper_energy_window=11.0 eV` to include the t_{2g} bands. Also, in this calculation the variable `flg_initial_guess_direc` has been set to 1. As a result, the display coordinate of the initial Gaussian is changed to the local coordinate system. With this flag set, in addition to the basic information of the initial Gaussian, the local coordinate information defining the display coordinates are additionally input. Specifically, it is input as follows.

```
dxy 0.20 0.50 0.50 0.50 -0.46 -0.46 0.76 0.46 0.46 0.76 -0.71 0.71 0.00
```

In the above, we place `dxy` Gaussian with width 0.2 (au) at the position of (0.50, 0.50, 0.50) in the lattice coordinates. Also, this Gaussian is displayed along the local coordinates ($\mathbf{L}_x, \mathbf{L}_y, \mathbf{L}_z$). Here, each local coordinate is as follows.

$$\begin{aligned}\mathbf{L}_x &= -0.46\mathbf{e}_x - 0.46\mathbf{e}_y + 0.76\mathbf{e}_z, \\ \mathbf{L}_y &= +0.46\mathbf{e}_x + 0.46\mathbf{e}_y + 0.76\mathbf{e}_z, \\ \mathbf{L}_z &= -0.71\mathbf{e}_x + 0.71\mathbf{e}_y + 0.00\mathbf{e}_z,\end{aligned}$$

Here, $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ represent the Cartesian coordinates which are used in the band calculation. The local coordinates $\mathbf{L}_x, \mathbf{L}_y, \mathbf{L}_z$ are defined on the basis of the TiO₆ octahedron unit. $\mathbf{L}_x, \mathbf{L}_y, \mathbf{L}_z$ and $bfe_x, \mathbf{e}_y, \mathbf{e}_z$ are shown in Fig. 19. Regarding the components of the local coordinates, the user must calculate the values from the structure information by themselves. In this case, we calculated them from the positions of the Ti atoms and the adjacent O atoms in the TiO₆ octahedron unit.

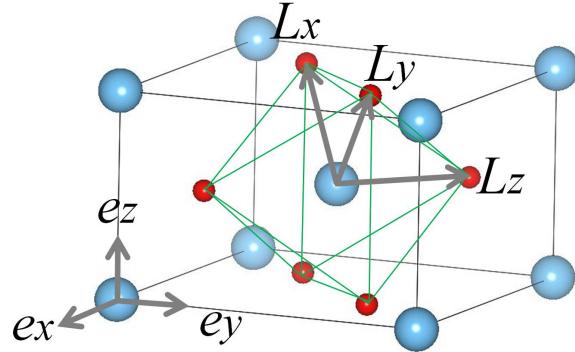


Figure 19: Definition of the local coordinates and the Cartesian coordinates in rutile type TiO_2 . The green frame describes TiO_6 octahedron unit.

Figure 20 is the resulting d_{xy} -type realspace Wannier function. It can be seen that the d_{xy} orbital is based on the local coordinates defined by Fig. 19.

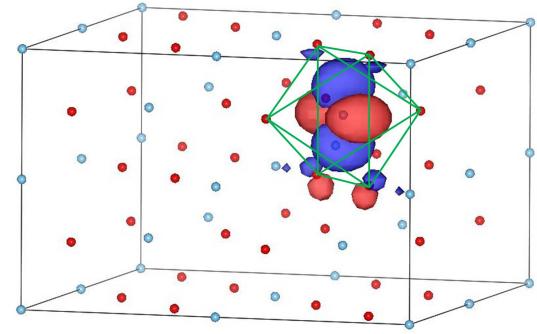


Figure 20: Calculated realspace t_{2g} Wannier function for rutile TiO_2 . The Wannier function data is superposed on the structure data whose size matches the display range of the Wannier function ($2 \times 2 \times 2$ supercell).

10 Si (Wannier function using B matrix)

This is an example for constructing the Wannier function by the linear combination of the initial Gaussians. We consider the Si calculation as an example. The following is `input.in` for this calculation. The band calculation is preformed using xTAPP with k -point sampling of $6\times 6\times 6$ (16 irreducible k points) and 36-Ry wavefunction cutoff. Note that the values of the energy window in `¶m_wannier` depend on the band-calculation software.

```

&param_wannier
N_wannier=8,                      !Lower limit of energy window (eV)
Upper_energy_window=17.0d0,          !Upper limit of energy window (eV)
N_initial_guess=8,                  !Number of initial Gaussian
flg_BMAT=1/                         !0:BMAT=unit matrix, 1:reading BMAT
s  0.5d0 0.00d0 0.00d0 0.00d0 !s@Si1
px 0.5d0 0.00d0 0.00d0 0.00d0 !px@Si1
py 0.5d0 0.00d0 0.00d0 0.00d0 !py@Si1
pz 0.5d0 0.00d0 0.00d0 0.00d0 !pz@Si1
s  0.5d0 0.25d0 0.25d0 0.25d0 !s@Si2
px 0.5d0 0.25d0 0.25d0 0.25d0 !px@Si2
py 0.5d0 0.25d0 0.25d0 0.25d0 !py@Si2
pz 0.5d0 0.25d0 0.25d0 0.25d0 !pz@Si2
0.50 -0.50 0.50 -0.50 0.00 0.00 0.00 0.00 !B_MAT(ig=1,iw=1:8)
0.50 0.50 -0.50 -0.50 0.00 0.00 0.00 0.00 !B_MAT(ig=2,iw=1:8)
0.50 -0.50 -0.50 0.50 0.00 0.00 0.00 0.00 !B_MAT(ig=3,iw=1:8)
0.50 0.50 0.50 0.50 0.00 0.00 0.00 0.00 !B_MAT(ig=4,iw=1:8)
0.00 0.00 0.00 0.50 0.50 -0.50 0.50 0.50 !B_MAT(ig=5,iw=1:8)
0.00 0.00 0.00 0.50 -0.50 0.50 0.50 0.50 !B_MAT(ig=6,iw=1:8)
0.00 0.00 0.00 0.50 0.50 0.50 -0.50 0.50 !B_MAT(ig=7,iw=1:8)
0.00 0.00 0.00 0.50 -0.50 -0.50 0.50 0.50 !B_MAT(ig=8,iw=1:8)
&param_interpolation
N_sym_points=5/ !Total number of symmetric k points in calculation lines
0.500 0.500 0.500 !Symmetric k point; SK_sym_pts(1:3,1); L
0.000 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); G
0.500 0.000 0.500 !Symmetric k point; SK_sym_pts(1:3,1); X
0.500 0.250 0.750 !Symmetric k point; SK_sym_pts(1:3,1); W
0.500 0.500 0.500 !Symmetric k point; SK_sym_pts(1:3,1); L
&param_visualization
flg_vis_wannier=1/ !Calculate realspace Wannier function (Do not: 0, do 1)

```

In this calculation, eight sp^3 -type Wannier functions of Si are constructed from a linear combination of eight s -type and p -type Gaussian functions. The B matrix information which are the coefficient of the linear combination is input, following the initial Gaussian information (look at the part with `B_MAT` tag). For the B matrix component, the user must calculate the values from the structure information by themselves. In this case, the components were calculated from the direction along the covalent bond of Si. The size of the B matrix is 8×8 , with rows representing the sequence of Gaussian functions and columns representing the sequence of the Wannier orbitals. The component B_{ij} of the B matrix is defined as follows.

$$w_j^{in}(\mathbf{r}) = \sum_i B_{ij} g_i(\mathbf{r}).$$

Here, g_i is the i th Gaussian function and w_j^{in} is the initial guess of the j th Wannier function. The input format of B matrix is as follows.

```

do ig=1,N_initial_guess
  read(5,*) (B_MAT(ig,iw),iw=1,N_wannier) !real(8)
enddo

```

If you set `reading_bmat_format=1`, you can switch in rows and columns of the B matrix. In this case, the row is the Wannier function and the column is the Gaussian function.

```

do iw=1,N_wannier
  read(5,*) (B_MAT(iw,ig),iw=1,N_initial_guess)
enddo

```

We show calculated sp^3 -type realspace Wannier function in Fig. 21.

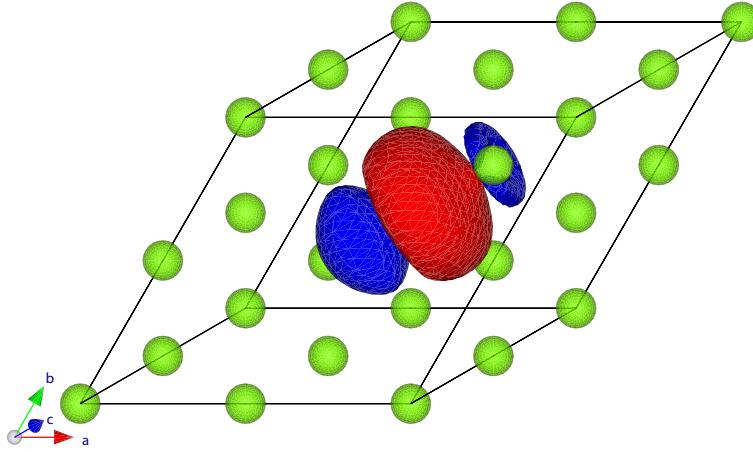


Figure 21: Calculated realspace sp^3 -type Wannier function of Si. The Wannier function data is superposed on the structure data whose size matches the display range of the Wannier function ($2 \times 2 \times 2$ supercell).

11 SrVO₃ (Interaction evaluation using constrained RPA)

Below is the input for computing the interaction parameters of the t_{2g} Wannier functions of SrVO₃ using constrained RPA.

```
&param_wannier
N_wannier=3,                      !Number of Wannier functions you want to calculate
Lower_energy_window=6.5d0,           !Lower limit of energy window (eV)
Upper_energy_window=9.7d0,           !Upper limit of energy window (eV)
N_initial_guess=3/                  !Number of initial Gaussian
dxy 0.5d0 0.5d0 0.5d0 !dxy@V
dyz 0.5d0 0.5d0 0.5d0 !dyz@V
dzx 0.5d0 0.5d0 0.5d0 !dzx@V
&param_interpolation
N_sym_points=5/ !Total number of symmetric k points in calculation lines
0.500 0.500 0.500 !Symmetric k point; SK_sym_pts(1:3,1); R
0.000 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); G
0.500 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); X
0.500 0.500 0.000 !Symmetric k point; SK_sym_pts(1:3,1); M
0.000 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); G
&param_visualization
/
&param_chiqw
Ecut_for_eps=5.0d0,!Cutoff of polarization function (Ry)
flg_cRPA=1/          !Constrained RPA option (0: normal RPA, 1: constrained RPA)
&param_calc_int
/
```

The band calculation is preformed using xTAPP with k -point sampling of $6 \times 6 \times 6$ (20 irreducible k points), 81-Ry wavefunction cutoff. The total number of bands is 50. Note that the values of the energy window in `¶m_wannier` depend on the band-calculation software. The condition of the band calculation is loose. If you want convergence, it is better to take $10 \times 10 \times 10$ k points, 100-Ry wavefunction cutoff, 15-Ry dielectric function cutoff, total number of bands around 100. As SrVO₃ has more than one plasmon excitation, it is better to keep the frequency grid as dense as possible. Specifically, it is advisable to set `Num_freq_grid=150`. Note that the transition metal has a strong localization and the quantitative accuracy of the interaction parameters may be affected by the pseudopotential. It is better to use a pseudopotential that is as stiff as possible (see Chapter 16).

12 (TMTSF)₂PF₆ (Interaction evaluation using constrained RPA))

Below is the input for calculating the interaction parameters of an organic compound (TMTSF)₂PF₆ using constrained RPA. As the Wannier function, we made HOMO-type orbital localized in organic molecules. The calculations were carried out in System B of Institute for Solid State Physics. It is a parallel job of 72 MPI processes; there are 12 communities and each community has 6 MPI processes. Also, 1 MPI consists of 24 threads. The calculation time was about 100 minutes. xTAPP was used for the band calculation. Note that the values of the energy window in `¶m_wannier` depend on the band-calculation software.

```
&param_wannier
N_wannier=2,                                !Number of Wannier functions you want to calculate
Lower_energy_window=2.7d0,                      !^^c2^^83GLower limit of energy window (eV)
Upper_energy_window=4.5d0,                      !Upper limit of energy window (eV)
FLG_BMAT=1,                                     !0:BMAT=unit matrix, 1:reading BMAT
N_initial_guess=6/                             !Number of initial Gaussian
px 0.20 0.26635 0.45950 0.44800 !px@central site of TMTSF molecule #1
py 0.20 0.26635 0.45950 0.44800 !py@central site of TMTSF molecule #1
pz 0.20 0.26635 0.45950 0.44800 !pz@central site of TMTSF molecule #1
px 0.20 0.73365 0.54050 0.55200 !px@central site of TMTSF molecule #2
py 0.20 0.73365 0.54050 0.55200 !py@central site of TMTSF molecule #2
0.9995851478 0.0000000000 !B_MAT(ig=1,iw=1:2)
0.0012540916 0.0000000000 !B_MAT(ig=2,iw=1:2)
-0.0287742877 0.0000000000 !B_MAT(ig=3,iw=1:2)
0.0000000000 0.9996715835 !B_MAT(ig=4,iw=1:2)
0.0000000000 -0.0140153561 !B_MAT(ig=5,iw=1:2)
0.0000000000 -0.0214544857 !B_MAT(ig=6,iw=1:2)
&param_interpolation
N_sym_points=5/    !Total number of symmetric k points in calculation lines
0.000 0.500 0.000 !Symmetric k point; SK_sym_pts(1:3,1); Y
0.000 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); G
0.500 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); X
0.500 0.500 0.000 !Symmetric k point; SK_sym_pts(1:3,1); M
0.000 0.000 0.000 !Symmetric k point; SK_sym_pts(1:3,1); G
&param_visualization
/
&param_chiqw
flg_cRPA=1,          !Constrained RPA option (0: normal RPA, 1: constrained RPA)
MPI_num_qcomm=12/   !Total number of MPI communities
&param_calc_int
/
```

13 Parallel calculation for q points (chiqw)

Polarization calculations can be parallelized for q points. Consider the case where the total number of MPI processes is 64, the parallel degree for the q points is 4, and the total number of the calculation q points is 20. The parallel degree on the q points is specified with the variable `MPI_num_qcomm` in `¶m_chiqw`. If this variable is 4, all the 64 MPI processes are divided into 4 communities, and each community consists of 16 MPI processes (Fig. 22). Since the total number of the q points is 20, one community performs the polarization calculation for five q points.

```
&param_chiqw
MPI_num_qcomm=4,           !Total number of MPI communities
MPI_num_proc_per_qcomm=16, !Total MPI per community
```

The job is executed as follows. The input command in sh system is as follows. Below, `-np 64` is an instruction to use the 64 MPI processes. Note that this specification differs depending on the processing system and/or machine. In RESPACK, when the total number of the MPI processes (`np`) is divided by `MPI_num_qcomm`, if the remainder is not zero, the program is stopped.

```
export OMP_NUM_THREADS=16
export MKL_NUM_THREADS=16
mpirun -np 64 ./calc_chiqw < input.in > LOG.chiqw
```

- ❑ Number of total MPI: 64 (`-np 64`)
- ❑ Number of community: 4 (`MPI_num_qcomm=4`)
- ❑ Number of MPI per community: 16 (= 64/4) (`MPI_num_proc_per_qcomm=16`)

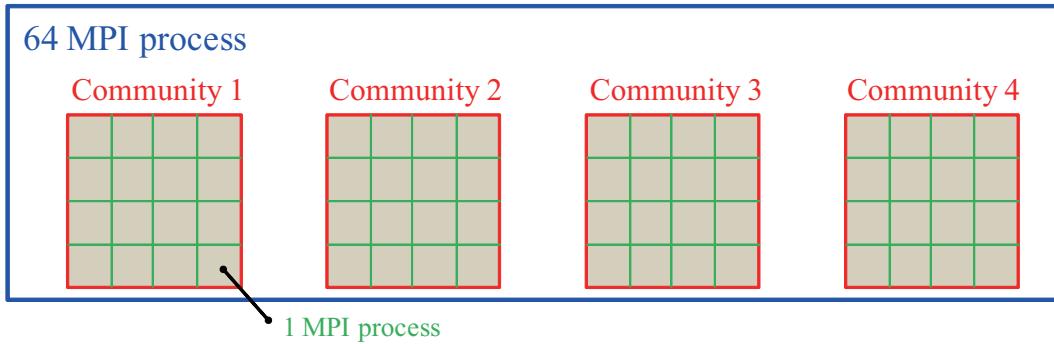


Figure 22: Schematic diagram of parallel computation on q points (the total number of MPI processes `np` is 64). By setting the q -point parallel degree to 4 `MPI_num_qcomm=4`, all the MPI processes are divided into 4 MPI communities. Each MPI community consists of 16 MPI processes. In RESPACK, when dividing the total number of MPI processes (`np`) by `MPI_num_qcomm`, if the remainder is not zero, the program is stopped.

14 Chiqw calculation for specific q points

Below is the input for executing the `chiqw` calculation for specific q points. In this case, we set the variable `flg_calc_type=2` and give the number of the q points we want to calculate (`n_calc_q=5`). Then, below the namelist, we specify the numbers of q points we want to compute; `4 8 14 22 27`. Note that this number is along the sequence of the `dat.sample-k` which contains irreducible k points. In this example, the five q points 4, 8, 14, 22, and 27 are calculated.

```
&param_chiqw
  flg_calc_type=2, !0:all-q-calc, 1:gamma-only, 2:bulkjob mode
  n_calc_q=5/      !Total number of q points to calculate
  4 8 14 22 27    !Numbers of q points to be calculated
```

15 Visualization for specific realspace Wannier orbitals

Below is the input for executing the visualization calculation for specific realspace Wannier functions. In this case, in the namelist `¶m_visualization`, we give the number of the Wannier functions we want to visualize (in this example, `N_write_wannier=3`). Then, below the namelist, we list the numbers of the Wannier functions we want to visualize. In this example, the 1st, 5th and 8th Wannier functions are visualized.

```
&param_visualization
  N_write_wannier=3/ !Number of realspace Wannier functions you want to calculate
    1 5 8           !Number of the realspace Wannier function you want to calculate
```

16 Pseudopotential dependence of interaction parameters

As an example, consider the pseudopotential dependence of the interaction parameter of SrVO₃. The pseudopotential depends on the local potential cutoff parameter r_c . The pseudopotential constructed with small r_c becomes deeper one, and the pseudo wavefunction obtained with this pseudopotential tends to distribute near the ion core. Figure 23 compares the pseudo wavefunctions with different cutoff r_c with the all-electron wavefunction (black thick line) for a vanadium atom. As the r_c is reduced to 2.1 Å(blue line) → 1.8 Å(green line) → 1.5 Å(red line), the maximum amplitude position of the pseudo wavefunction is shifted to the ion-core side. This tendency is noticeable in orbitals without nodes like the 1s, 2p, and 3d orbitals. The pseudopotentials with r_c = 1.8 Åand 2.1 Åare ionized valence configurations ($3d^34s^04p^0$), and the pseudopotential with r_c = 1.5 Åwas created for the ionized semicore configuration ($3s^23p^63d^3$). Also, the Troullier-Martins (TM) type was adopted as a function form of the pseudo wavefunction.

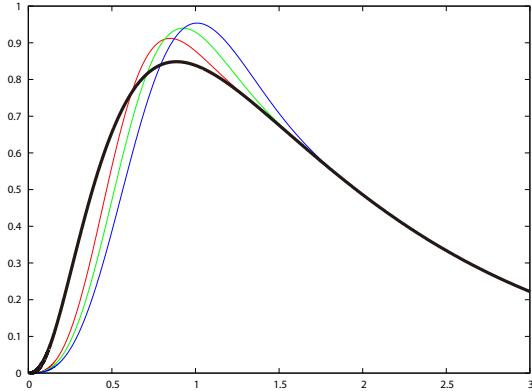


Figure 23: Comparison of the radial part of the pseudo wavefunction and the all-electron wavefunction (black thick line) of the 3d orbital of a vanadium atom. The figure shows that, as the local potential cutoff parameter r_c changes to 1.5 Å(red line), 1.8 Å(green line), 2.1 Å(blue line), how the resulting pseudo wavefunction changes. The horizontal axis is the distance r (Å). By reducing the r_c parameter, the maximum amplitude position of the pseudo wavefunction is shifted to the ion-core side ($r = 0$).

This tendency can also be seen in the Wannier function. The Wannier function with a pseudopotential with a small r_c tends to be more localized. As a result, it gives a large direct integral. Table 1 is the r_c dependence of interaction parameters for the t_{2g} Wannier function of SrVO₃. The calculation conditions are a $6 \times 6 \times 6$ k -point sampling, a wavefunction cutoff of 144 Ry. Dielectric-function cutoff is 5 Ry, and the total number of bands is 100. We see from the table that r_c and the Wannier spread Ω has positive correlation, where Ω is defined as

$$\Omega = \sqrt{\langle r^2 \rangle - |\langle \mathbf{r} \rangle|^2}.$$

It can also be seen that the r_c and onsite direct integrals U , U' have a negative correlation. There is no noticeable effect on the exchange integral J and the nearest neighbor direct integral V . As pseudopotentials, besides the TM type, there are the RRKJ (Rappe-Rabe-Kaxiras-Joannopoulos) type and the ONCV (Optimized Norm-Conserving Vanderbilt) type. Even if the same cutoff r_c is employed, the results can be quantitatively different due to the difference in the functional form of the pseudopotential, so be careful.

Table 1: Pseudopotential dependence of interaction parameters of SrVO₃. “bare” and “cRPA” represent bare and screened Coulomb interactions. U , U' , J , and V are onsite intraorbital direct integral, onsite interorbital direct integral, onsite exchange integral, and nearest intraorbital direct integral, respectively. The unit of the local potential cutoff r_c in the pseudopotential and the Wannier spread Ω are Å. The unit of interaction is eV.

| - | - | bare | | | | cRPA | | | |
|-------|----------|-------|-------|------|------|------|------|------|------|
| r_c | Ω | U | U' | J | V | U | U' | J | V |
| 1.5 | 1.354 | 15.86 | 14.60 | 0.59 | 3.74 | 3.65 | 2.49 | 0.56 | 0.67 |
| 1.8 | 1.377 | 15.54 | 14.26 | 0.60 | 3.74 | 3.54 | 2.36 | 0.57 | 0.70 |
| 2.1 | 1.382 | 15.15 | 13.87 | 0.60 | 3.74 | 3.50 | 2.32 | 0.57 | 0.72 |

17 File format (reading format)

1. dat.bandcalc (Band calculation information)

```
OPEN(117,FILE='./dir-wfn/dat.bandcalc')
read(117,*) Ecutforspsi !Wavefunction cutoff energy (Ry) real(8)
read(117,*) FermiEnergy !Fermi energy (au) real(8)
read(117,*) Etot !Total energy (au) real(8)
```

2. dat.sample-k [Sample k point vector (irreducible k point)]

```
OPEN(101,FILE='./dir-wfn/dat.sample-k')
read(101,*) Nk_irr !Total number of irreducible k point integer
do ik=1,Nk_irr
  read(101,*) (SKI(i,ik),i=1,3) !Irreducible k point vector real(8)
enddo
```

3. dat.eigenvalue [Band energy (irrelevant k points)]

```
OPEN(111,FILE='./dir-wfn/dat.eigenvalue')
rewind(111)
read(111,*) NBAND !Number of bands
do ik=1,Nk_irr
  do ib=1,NBAND
    read(111,*) E_EIGI(ib,ik) !Band energy (au) (au) real(8)
  enddo
enddo
```

4. dat.nkm [Expanded G number (irreducible k point)]]

```
OPEN(132,FILE='./dir-wfn/dat.nkm-nw')
rewind(132)
do ik=1,Nk_irr
  read(132,*) NGI(ik) !Expanded G number integer
enddo
```

5. dat.kg [Expanded G vector (irreducible k point)]

```
OPEN(104,FILE='./dir-wfn/dat.kg')
do ik=1,Nk_irr
  read(104,*) NG_for_psi !Expansion G number at ik point integer
  do ig=1,NG_for_psi
    read(104,*) (KGI(i,ig,ik),i=1,3) !Expansion G vector at irreducible ik point integer
  enddo
enddo
```

6. `dat.wfn` [Wavefunction (irreducible k point)]

```
OPEN(102,FILE='./dir-wfn/dat.wfn',FORM='unformatted')
read(102) ncomp !Number of components
do ik=1,Nk_irr
  do ib=1,NBAND
    read(102)((C0(ic,i,ib,ik),ic=1,ncomp),i=1,NGI(ik))!Wave function complex (8)
  enddo
enddo
!In the current version, only ncomp = 1 is supported
```

7. `dat.lattice` (Lattice vector)

```
OPEN(105,FILE='./dir-wfn/dat.lattice')
read(105,*) a1(1), a1(2), a1(3) !a1 vector in bohr units real(8)
read(105,*) a2(1), a2(2), a2(3) !a2 vector in bohr units real(8)
read(105,*) a3(1), a3(2), a3(3) !a3 vector in bohr units real(8)
```

8. `dat.symmetry` (Symmetry, ++Expression that works in inverse space++)

```
OPEN(100,FILE='./dir-wfn/dat.symmetry')
read(100,*) nsymq !Number of symmetry operation integer
read(100,*) nnp !Partial translation integer
do iop=1,nsymq
  read(100,*) ((rg(i,j,iop),i=1,3),j=1,3) !Rotation symmetry integer
  read(100,*) (pg(i,iop),i=1,3)           !Partial translational symmetry integer
enddo
```

9. `dat.atom_position` (Atomic position)

```
OPEN(108,FILE='./dir-wfn/dat.atom_position')
read(108,*) nsi !Number of atoms
do i=1,nsi
  read(108,*) chemical_species(i), !Atomic species character
+            (asi(j,i),j=1,3)      !Atomic position real(8)
enddo
```

18 License

The source codes and utility programs and scripts of this software are distributed on the basis of GNU General Public License version 3 (GPL v3).

19 How to quote

When quoting RESPACK, it would be appreciated if you could quote the following papers:

1. K. Nakamura, Y. Nohara, Y. Yoshimoto, Y. Nomura, Phys. Rev. B **93**, 085124 (2016).
2. K. Nakamura, Y. Yoshimoto, T. Kosugi, R. Arita, M. Imada, J. Phys. Soc. Jpn **78**, 083710 (2009).
3. K. Nakamura, R. Arita, M. Imada, J. Phys. Soc. Jpn **77**, 093711 (2008).
4. Y. Nohara, S. Yamamoto, T. Fujiwara, Phys. Rev. B **79**, 195110 (2009).
5. T. Fujiwara, S. Yamamoto, Y. Ishii, J. Phys. Soc. Jpn. **72**, 777 (2003).