Dysurf: A program for simulating four-dimensional dynamical structure factors

Changpeng Lin
changpeng.lin@epfl.ch

Jiawang Hong
hongjw@bit.edu.cn

CONTENTS

June, 2021

1 Compile and use Dysurf program

The Dysurf program is mainly written in Fortran 90 and use some Fortran 2003 extensions. To compile the Dysurf program, one need to go to the *src* subdirectory of the distribution and modify the Makefile according to personal environment settings. The Dysurf program only depends on external Linear Algebra PACKag (LAPACK) and please provide the root where LAPACK library is installed either by specifying $LDFLAGS or $LAPACK. Intel Fortran compiler is recommended for compiling this program with convenient Math Kernel Library (MKL).

After complication succeeds, a Dysurf executable will be created in the same directory. To run the Dysurf program, use the command line as

./dysurf *input_file_name*

Any filename of the input is acceptable by the Dysurf program. Then, the Dysurf program will read the settings in input file and run a calculation.

2 Input files

There are two files required to run the Dysurf program. One is the aforementioned input file containing computational settings, and the other is second-order force constants provided either in PHONOPY (http://atztogo.github.io/phonopy/) format or Quantum ESPRESSO (http://www.quantum-espresso.org/) q2r.x format. The filename of force constants must be FORCE_CONSTANTS and espresso.fc, respectively.

The input file controlling the whole calculation consists of two namelists and three bodies, LATTICE_PARAMETERS, ATOMIC_POSITIONS and BORN. Some tags are mandatory, whereas others are optional and take a default value when unspecified. You can find the example input file in *examples* subdirectory.

- ***&basic* namelist**

ntypes (integer, mandatory): number of different atomic types in the given system

natoms (integer, mandatory): total number of atoms in the given system

nsize (integer, 3, mandatory): supercell sizes along three crystal axes used for force constant calculation

- ***&inputsqe* namelist**

nat (integer, *ntypes*, mandatory): number of atoms in each atomic type

elements (string, *ntypes*, mandatory): name of each atomic type

clatvec (real, 3×3, optional): lattice vectors for conventional unit cell in angstrom, must be specified when

primitive unit cell is not orthorhombic.

ne (integer, default = 1000): number of points of phonon energy, E_{max} = *ne·deltaE*

deltaE (real, default = 0.01): phonon energy resolution, E_{max} = *ne·deltaE*

nqh (integer, default = 100): number of q-points along the calculated dispersion path

nqk (integer, default = 6): number of q-points along one path perpendicular to the calculated dispersion path, and the other path in case of TDS calculation

nql (integer, default = 6): number of q-points along the other path perpendicular to the calculated dispersion path

deltaH (real, default = 0.01): wavenumber resolution along the calculated dispersion path in reciprocal lattice basis of orthogonalized conventional unit cell

deltaK (real, default = 0.005): wavenumber resolution along one path perpendicular to the calculated dispersion path in reciprocal lattice basis of orthogonalized conventional unit cell

deltaL (real, default = 0.005): wavenumber resolution along the other path perpendicular to the calculated dispersion path in reciprocal lattice basis of orthogonalized conventional unit cell

q0 (real, 3, mandatory): the origin of dispersion path in reciprocal lattice basis of orthogonalized conventional unit cell

path (real, 3×3, mandatory): Only *path(:,1)* is mandatory, which is the phonon dispersion path in crystalline direction (orthogonalized conventional unit cell basis) used in calculations, the end q-point of dispersion is determined by *[q0(1)+deltaH·nqh·path(1,1), q0(2)+deltaK·nqk·path(2,1), q0(3)+deltaL·nql·path(3,1)]*. The *path(:,2)* and *path(:,3)* are optional, corresponding to the other two directions perpendicular to the calculated dispersion path

masses (real, *ntypes*, default = automatic): atomic masses corresponding to each atomic type. If they are omitted, the program can automatically research their atomic masses by element name.

coh_b (real, *ntypes*, default = automatic): coherent scattering length of neutron for each element. If omitted, the program can automatically fill their values by element name.

xray_b (real, *ntypes*, default = automatic): scattering length of X-ray (approximate as atomic number) for each element. If omitted, the program can automatically fill their values by element name.

temp (real, default = 300.0): temperature (K) used in calculations

temp_min (real, default = 100.0): minimum temperature (K) used in calculations

temp_max (real, default = 1000.0): maximum temperature (K) used in calculations

temp_step (real, default = 0.0): temperature step (K) between *temp_min* and *temp_max*, if the default 0.0 is used, the calculation is only performed once at *temp*.

qmesh (integer, 3, default = /20, 20, 20/): phonon mesh used in root square displacement (RMSD) calculation

nonanalytic (logical, default = .FALSE.): if .TRUE., compute the nonanalytic part of dynamical matrix to LO-TO splitting in polar materials, Born effective charge and dielectric constant should be provided in BORN body

espresso (logical, default=.FALSE.): if .TRUE., read the second-order force constants in Quantum Espresso q2r.x format from the espresso.fc.

lneutron (logical, default = .TRUE.): if .TRUE., calculate the dynamic structure factor of inelastic neutron scattering, *lneutron* and *lxray* must be one .TRUE. and one .FALSE.

lxray (logical, default = .FALSE.): if .TRUE., calculate the dynamic structure factor of inelastic X-ray scattering, *lneutron* and *lxray* must be one .TRUE. and one .FALSE.

ltds (logical, default = .FALSE.): if .TRUE., calculate the thermal diffuse scattering (TDS)

lphase (logical, default=.FALSE.): if .FALSE. discard the phase from the dot product between momentum transfer and atomic positions in unit cell

write_rmsd (logical, default = .TRUE.): if .TRUE., write RMSDs into file, the filename is set by *filename_rmsd*

read_rmsd (logical, default = .FALSE.): if .TRUE., read RMSDs from file, the filename is set by *filename_rmsd*

filename_rmsd (string, default = "rmsd.dat"): filename of RMSD to write or read

filename_omega (string, default = "omega.dat"): filename of phonon energy to write

lresfunc (logical, default = .FALSE.): if .TRUE., consider the Gaussian instrument resolution function, only energy resolution is implemented, the type of resolution function is specified by *functype*

functype (string, default = "CNCS12"): type of Gaussian instrument resolution function used in calculations, it can be "CNCS12", "CNCS20" and "poly".

order (integer, default = 4): the order of polynomial, if *functype* is "poly", user-defined polynomial function is used to determine the standard deviation of a given phonon energy

paras (real, *order*+1, no default): parameters of polynomial if *functype* is "poly", e.g. when *order* = 4, paras will be a one-dimensional array with 5 elements

xm (integer, default = 1): enlarge the energy standard deviation by *xm* times

degauss (real, default = 0.5): the broadening in meV of Gaussian smearing, if *lresfunc* is .FALSE., normal Gaussian function is used to implement energy conservation during scattering events

aff_wk (logical, default = .FALSE.): if .TRUE., the scattering length of X-ray for each element determined by atomic form factor (a linear combination of five Gaussians, see *D. Waasmaier and A. Kirfel, Acta Cryst. A51, 416, 1995* for details), this can be more accurate than using atomic number approximation.

aff_a (real, 5×ntypes, no default): the a_i parameters entering the atomic form factor and five values (a_1, a_2, …, a_5) for each element, only effective when *aff_wk* = .TRUE.

aff_b (real, 5×ntypes, no default): the b_i parameters entering the atomic form factor and five values (b_1, b_2, …, b_5) for each element, only effective when *aff_wk* = .TRUE.

aff_c (real, ntypes, no default): the *c* parameters entering the atomic form factor and one value for each element, only effective when *aff_wk* = .TRUE.

- **Body of LATTICE_PARAMETERS**

This body gives the lattice parameters of the system. It has 5 lines with the first line entered as the body name "LATTICE_PARAMETERS", followed by a real number called scaling factor (the same meaning of the second line of POSCAR of VASP program). The third to fifth lines is specified by three lattice vectors in angstrom. An example is shown as

```
LATTICE_PARAMETERS
1.0
      3.9535210380492276     -0.0000000000000000     -0.0000000000000000
     -0.0000000000000000      3.9535210380492276     -0.0000000000000000
     -0.0000000000000000     -0.0000000000000000      9.0753729051409913
```

- **Body of ATOMIC_POSITIONS**

This body should provide the atomic positions of *natoms* atoms either in reduced lattice coordinate or Cartesian coordinate. The first line of this body is the body name "ATOMIC_POSITIONS" and the second line gives which coordinate is used. Any word starting as "D" or "d" will be interpreted as *direct* and reduced lattice coordinate is used. Besides, any word starting as "C" or "c" will be interpreted as *Cartesian* and Cartesian coordinate is then used in angstrom. The following *natoms* lines are atomic positions, which should be in the same atomic type order in *elements*, and an example of this body is

```
ATOMIC_POSITIONS
direct
   0.2500000000000000   0.2500000000000000   0.1396757182675747
   0.7500000000000000   0.7500000000000000   0.8603242527324266
   0.7500000000000000   0.2500000000000000   0.5000000000000000
   0.2500000000000000   0.7500000000000000   0.5000000000000000
   0.2500000000000000   0.2500000000000000   0.6730958609573343
```

0.7500000000000000 0.7500000000000000 0.3269041390426587
0.7500000000000000 0.2500000000000000 0.0000000000000000
0.2500000000000000 0.7500000000000000 0.0000000000000000

- **Body of BORN**

This body is used only when *nonanalytic* is set as .TRUE. to consider the nonanalytic part of the dynamical matrix. The first line is the body name "BORN", which is followed by dielectric constants (3 lines) and then Born effective charges (3×*natoms* in total). An example of this body for a unit cell with 2 atoms is shown as

BORN
6.358477 0.000000 0.000000
0.000000 6.358477 0.000000
0.000000 -0.000000 6.358477
2.55348 0.00000 -0.00000
0.00000 2.55348 -0.00000
0.00000 -0.00000 2.55348
7.38609 0.00000 -0.00000
0.00000 7.38609 0.00000
0.00000 0.00000 7.38609
-2.03798 0.00000 0.00000
0.00000 -5.86515 0.00000
0.00000 -0.00000 -2.03798
-2.03798 0.00000 0.00000
0.00000 -2.03798 0.00000
0.00000 -0.00000 -5.86515
-5.86515 0.00000 0.00000
0.00000 -2.03798 0.00000
0.00000 -0.00000 -2.03798

3 Output files

The Dysurf program have the following output files.

- **SQE_*$temp*.dat**: output files containing information of the calculated SQE, which have *nqh* rows and *ne* columns (*nqh* rows and *nqk* columns in case of TDS calculations). The distances from origin of dispersion path for each q-point is not written because it can be set in the process of plotting SQE. *$temp* is the aim temperature, if there are more than one temperature used for SQE calculations.
- **qpoints.dat**: output file containing the list of qpoints along the dispersion path, which has *nqh* rows and 4 columns. The first column is just the index for each qpoint from 1 to *nqh*, and the remaining 3 columns are the corresponding qpoint in the basis of reciprocal lattice of primitive unit cell.
- **omega.dat**: output file containing information of phonon energy in meV, which has *nqh* rows and

nbands+1 columns. The first column is the distance away from the origin along the dispersion path, and the remaining columns are phonon energy for each band.

- **rmsd.dat**: output file containing information of RMSD for each temperature and each atom along three Cartesian axis. An example output of rmsd.dat for BiCuSeO is shown as

```
300.00
Bi        0.10354992      0.10354992      0.09820327
Bi        0.10354992      0.10354992      0.09820327
Cu        0.14973965      0.14973965      0.13318656
Cu        0.14973965      0.14973965      0.13318656
Se        0.10958776      0.10958776      0.10239609
Se        0.10958776      0.10958776      0.10239609
O         0.09235377      0.09235377      0.10096111
O         0.09235377      0.09235377      0.10096111
```

The first line is temperature in kelvin, and the following *natoms* lines are atomic type, RMSDs in angstrom along three Cartesian axis, respectively, for each line. If more than one temperature is used, other data blocks like this for each temperature will be written.

4 Additional comments

1) The subroutine used to build and solve dynamical matrix of phonons is adapted from ShengBTE, in which non-analytic correction to treat the LO-TO splitting at the Brillouin zone center has some problems. No correction is applied exactly at the Γ point in order not to rely on guesses about directions. This sometimes may lead to unphysical behaviors of phonon branches near the Γ point, but this should not affect SQE results.

2) The current version of Dysurf program does not perform any acoustic sum rule (ASR) corrections to force constants as well as symmetrization of Born effective charges and dielectric constants, which are both very important for determining the correct phonon band structures. You need fulfill them by other modelling packages like PHONOPY and the PHONON package of Quantum ESPRESSO.

3) Please take care of some elements that has imaginary part of neutron scattering length, which means there may be a strong adsorption of neutrons by these elements. Although our current Dysurf program has included the first 96 elements in periodic table and you usually do not need to manually set the scattering lengths of neutron (*coh_b*) and X-ray (*xray_b*), it is highly recommended to check whether the scattering length is specified correctly, especially for Cd, Sm, Gd, Dy, Po, At, Rn, Fr, Ra, Ac, Pu and Cm.

4) The most expensive part of Dysurf program is to calculate RMSDs which need somewhat dense phonon mesh to obtain converged results and must be done in the full Brillouin zone. You can save time after the first calculation by setting *write_rmsd* = .TRUE. to write RMSDs into file and then set *read_rmsd* = .TRUE. when performing other calculations on the same system.

5) The FORCE_CONSTANTS file of latest PHONOPY has changed a little in term of its format, i.e. the first line of this file. The new format contains two numbers in the first line, however the old one only has a number (the total number of atoms in the supercell). The Dysurf program follows the old format and please edit accordingly if this issue rise.