Tutorial

(Frozen-phonon method for state anticrossing situations and its application to zero-point motion effects in diamondoids.)

Pablo García-Risueño[1,2], Peng Han[1,3], Surender Kumar[1], and Gabriel Bester[1,2]

[1]Institut für Physikalische Chemie, Universität Hamburg, Grindelallee 117, D-20146 Hamburg, Germany
[2]The Hamburg Centre for Ultrafast Imaging, Luruper Chaussee 149, D-22761 Hamburg, Germany
[3]Department of Physics, Capital Normal University, Beijing Key Lab for Metamaterials and Devices, Beijing 100048, China

February 9, 2022

In the present tutorial, we explain the calculation steps for the frozen-phonon renormalization of electronic eigenvalues of nanostructured (non-bulk) systems, avoiding errors due to the possible crossing and anticrossing of states (See Ref [1]). The codes involved in the calculations are:

- **Quantum Espresso** (executables are **pw.x, ph.x, dynmat.x**)**:** Structure optimisation and SCF & phonon calculations (input files are **geomopt.in, scf.in, dyneq.in, dymat.in**). We provide the files for QE, but any other DFT code could be used if the files are (slightly) adapted.
- **vib_pot.f90 :** To generate the input files with displaced nuclear positions (input file **vib_pot.in**).
- **frozen-phonon.f90 :** Evaluation of the first approximation to the renormalization of electronic eigenvalues (input file **frozenphonon.in**).
- **FPDoubleAnticrossing.f90 :** For the treatment of the anticrossing of two states (input file (**FPanticrossing.in.**)
- **FPTripleAnticrossing.f90 :** For the treatment of the anticrossing of three states (input file (**FPanticrossing.in.**)

Sample input files are available at [3].
There are six calculation steps, explained in the following.

1 Geometry optimization (relaxation)

- Using QE (any other DFT code would do, but our sample files are for QE) optimize the structure geometry (input file: `geomopt.in`) in a cubic supercell (hardwired to a cubic supercell) until you reach `Total force` in the order of $< 10^{-6}$ Ha/Bohr.

2 Calculation of the phonon frequencies and vibrational eigenvectors

- With QE, perform a ground state calculation with relaxed geometry.
- Solve the dynamical equation using `ph.x` with input file `dyneq.in`. You will get vibrational frequencies file `DynG` (without non-analytical part (see [4])) and vibrational wavefunction files (XXX.wfcphononmode i.e XXX.wfc7, here 7 stands from 7th phonon mode and XXX stands for prefix in QE input file).

- Add the non-analytical part to the frequencies (by using certain acoustic sum rules) by running the executable `dynmat.x` with the input file `dynmat.in`.
- Run the script `run_dyneq.sh` (adapted to your system). The output file:
  - `dynmat.out`: contain phonon frequencies and eigenmodes which will be used for further calculations.

3 Generation of files with displaced nuclear positions for the finite-difference (i.e. frozen-phonon) calculations.

The programm `vib_pot.f90` creates several `displaced-modeXX_scf.in` (QE input files, XX stands for phonon mode) with nuclear positions displaced through the normal modes. The program `vib_pot.f90` creates directory frozen-R & frozen+R and puts all the `displaced-modeXX_scf.in` files there.

- Compile the code, for instance in case of Intel compiler:

```
$ ifort -o vib_pot.x vib_pot.f90
```

- Run the programm `vib_pot.x` with input file `vib_pot.in`:

Here is a sample vib_pot.in :

```
-------------------------------------------------------------------------------

QEMassesFile = out_scf.out         # QE output file from SCF
InputQEscffile = scf.in            # QE input file for SCF
InputQEpotfile = Veff-unperturb.in # Input file that you don't need to change
FreqsUfile = dynmat.out            # file with Phonon frequency and eigenmodes
BasicDirPath = ./.                 # path of your directory
Finite_Difference_Parameter = 4.0  # h in a.u.
%Atoms=
C  12.0107      14      # atom name,  atomic mass, total number of C in  system.
H  1.007825035  20      # atom name,  atomic mass, total number of H in  system.
%

-------------------------------------------------------------------------------
```

The required input files, along `vib_pot.in` are:

- `Veff-unperturb.in`. Does not need to be modified (plot utility input).

Here is the file Veff-unperturb.in :

```
-------------------------------------------------------------------------------
&inputPP
prefix='diam-',
outdir='./',
plot_num=1,
spin_component=0,
filplot = 'Veff_unperturb.pot'
/
&plot
iflag=3,
output_format=5
/
-------------------------------------------------------------------------------
```

- `scf.in`, QE input file with undistorted positions for SCF. Make sure that `startingwfc='atomic'` is present in this file.
- Note: We recommend to use a value of `Finite Difference Parameter(h)`=4.0 in `vib_pot.in`, but the user can freely choose.
- WARNING: Make sure that the `Finite Difference Parameter(h)` of `vib_pot.in` is the same as that of `frozenphonon.in` (see later), otherwise the calculated renormalizations would be wrong.

You will get one more output file in your current directory:

- `Uvec.dat`: Phonon frequency and Eigenvectors (in atomic unit)

4 Calculation of undisplaced and displaced electronic eigenvalues and wavefunctions

We are going to perform these calculations using scripts which can be run in parallel.

- Copy the files `run-wfcs0.sh`, `run-wfcs-a.sh`, `run-wfcs-b.sh`, `run-wfcs+a.sh` and `run-wfcs+b.sh` to your working directory.
- Modify these script files according to your computing facility (adjust paths).
- Copy `wfcs.in` to your working directory (the file will not be modified).
- Run the script files. The script files will:
  - Create individual directories for each eigenmodes and run the QE calculations there. The directories are removed after completion and the relevant output (`eigenval-modeXX.xml`, `out_scf-modeXX.out`) are copied to `outputfiles/outputsVeffcalc/displaced-` and `outputfiles/outputsVeffcalc/displaced+` ;
  - Copy wavefunction files for undisplaced nuclear positions X_0 called `wfcYY.wfn`(YY stands for electronic states) in the `Wfcs/Wfcs-undisplaced/` folder;
  - Copy wavefunction files for $X_0 \pm h \cdot U_\nu$ called `modeXX-wfcYY.wfn` (`XX` stands for phonon mode and `YY` stands for electronic state) to the `Wfcs/Wfcs-displaced+` and `Wfcs/Wfcsdisplaced-` folders respectively.

5 Evaluation of the renormalization of electronic eigenvalues in the absence of anticrossings

To perform this calculation, you will need the file `frozen-phonon.f90` that will be subsequently compiled and that will read the input files

- `frozenphonon.in`

```
    Here is the sample frozenphonon.in .
-------------------------------------------------------------------------------
!Initial_Temperature = 0.0001d0 K  #only if temperature dependence is studied.
!Final_Temperature = 350.0 K       #only if temperature dependence is studied.
!Number_of_temperatures = 4        #only if temperature dependence is studied.
Temperature = 0.0000001 K          #set (nearly) 0 for zero point renormalization.
Number_of_atoms = 26               #Number of atoms in system.
Number_of_species = 2              #Type of atoms in system.
elec_eigval_file_name = out_scf0.out   #output file for undisplaced modes.
xml_eigval_$file_name = eigenval.xml   #eigenvalue file from QE.
Band_occupation = 2        # 1 for spin-orbit; 2 without spin-orbit
```

```
Finite_Difference_Parameter=4.000000   #displacement parameter.
phonon_frequency_file_name = Uvec.dat  #file with phonon frequencies.
phonon_frequency_units=cm^{-1}         # units of the phonon frequencies
Check_crossover = 0                    # 1: crossover check ; 0: no check
Crossovers_from_file = 0  # Name of wavefunction file for crossover check.
InitialStateFPh = 26      # Initial state
FinalStateFPh = 28        # Final state
InitialMode = 7           # Initial phonon mode
FinalMode = 78            # Final phonon mode
HOMO_imposed_degeneracy = 3      # impose degeneracy of HOMO-states
LUMO_imposed_degeneracy = 1      # impose degeneracy of LUMO-states
Eigenvalues_from_xml_files = 1   # option to choose eigenval.xml file
folder_output_files_displaced+ =outputfiles/outputsVeffcalc/displaced+ # files path
folder_output_files_displaced- =outputfiles/outputsVeffcalc/displaced- # files path
-------------------------------------------------------------------------------
```

- `out_scf0.out` : output file for undisplaced modes.
- `eigenval.xml`: eigenvalue file from QE for undisplaced modes in .xml format.
- `Uvec.dat`: file with phonon frequencies and eigenvectors (in atomic units).
- `run-FF-withcrossovers.sh` (this script compiles and runs frozen-phonon.f90).

The actual calculations of the renormalisation goes as follows:

- (Optional, this is not necessary but can deal as quick check). Perform a calculation to obtain a first approximation of the frozen-phonon renormalization by setting `Check_crossover = 0` in `frozenphonon.in` (this will not take into account possible state crossings and consequently not calculate overlaps) by running the script as:

  ```
  $./run-FF-withcrossovers.sh
  ```

  (this compiles frozen-phonon.f90 and runs it).

- Set `Check_crossover = 1` in `frozenphonon.in` to obtain results, where the effect of crossings is removed (the effect of anticrossings might still need to be corrected, see next section). Adjust the paths for your system and run `run-FF-withcrossovers.sh`. This calculation is more time-consuming because it calculates wavefunction overlaps. For the state to renormalized (e.g. the HOMO) the program calculates the overlap between i) the (HOMO) wavefunction calculated with undisplaced nuclear positions, and ii) the wavefunctions of nearby states (e.g. HOMO, HOMO-1 and HOMO-2) evaluated with displaced nuclear positions (displaced through the directions of the eigenvectors, $\pm U_\nu$, of every vibrational mode). This allows to "follow" a certain state through a crossing (the overlap between the state calculated with undisplaced coordinates and the one calculated with displaced coordinates will be high), making sure the eigenvalues in the eigenvalue vs. h file all belong to the same state.

A file called `out-FF-withcrossovers.out` (in both cases) will appear after the run with the values of the frozen-phonon renormalization ("difference") for HOMO and LUMO.

6 Correction to the renormalization in case of anticrossings

- **Check if there is potential distortion due to anticrossings:**
  This is done by looking at the overlap of wavefunctions in the output file of the calculations made with `Check_crossover = 1` in frozenphonon.in. Our experience shows that a value of the overlap below 0.9975 signifies the presence of an anticrossing (overlaps without anticrossing are around 0.999) that must be corrected using the procedure that we explain below. These modes usually show also a ZPM contribution that deviates significantly from all the other modes.

- **Rerun calculations (`vib_pot.x`) for different values of h:** Just for the vibrational modes which indicated a possible anticrossing, repeat the calculations of eigenvalues for different values of h (e.g.$\pm$1, $\pm$2, $\pm$3, $\pm$4, $\pm$5, $\pm$6, $\pm$8, $\pm$10, $\pm$16).

  This is done by setting the `Finite_Difference_Parameter` variable to 1.0, 2.0 etc. in `vib_pot.in`.

- **Arrangement of the data:** In a directory which contains the files called `Uvec.dat`, `eigenval.xml` and `out_scf.out`, as well as the directories `displaced+` and `displaced-` with the eigenvalue files, compile (e.g. with gfortran here) `GenerateDataToPlotCrossovers.f90`:

```
gfortran -o GenerateDataToPlotCrossovers.x GenerateDataToPlotCrossovers.f90
```

  and then run

```
./GenerateDataToPlotCrossovers.x
```

  The input file is `GenerateDataToPlotCrossovers.in`; an example of it is:

```
Number_of_species = 2   ! Number of different atomic numbers at the system
Read_from_xml_file = 1  ! Keep to "1" (.xml file is read)
elec_eigval_file_name = out_scf.out  ! Output of Quantum Espresso
Band_occupation = 2  ! Number of electrons for every given eigenvalue  (always 2)
Modes_to_analyse###### ! Keep this keyword exaclty like this
46                    ! Indices of modes where an anticrossing is expected.
47
53
54
###################            ! Keep these hashes here (delimiters)
Displacement_parameters######  ! Keep this keyword exaclty like this
01                             ! Finite difference parameter h (a.u.) for each mode
02
03
05
06
07
08
10
16
############################
```

  This code outputs files which include the eigenvalues of a certain state as a function of the displacement parameter h. For each vibrational mode and each electronic eigenvalue (HOMO, LUMO, etc...) a file is created. So a potentially large number of of (N_eigenvalues $\times$ N_vib.modes) files will be created. As illustration: In Fig. 1 the collected results for three states for vibrational mode 56 are shown. On the left part of Fig. 1, the grey, orange and blue datapoints would be each in one individual file (3 files). In this case, some manual rearrangement is necessary to obtain a dataset that corresponds to the right side of Fig. 1 where the datapoints a meaningfully assigned to the states. So you will have to exchange datapoints by hand between two files to finally have two files that contain smooth curves that correspond to the right of Fig. 1 (on the left of Fig. 1 the orange and blue datasets show a kink, which we have to avoid).

- **Actual calculation of the correction:** Make a visual inspection of the shape of the curves of the eigenvalues vs h (`ToPlot/modeXX/stateYY.out`,

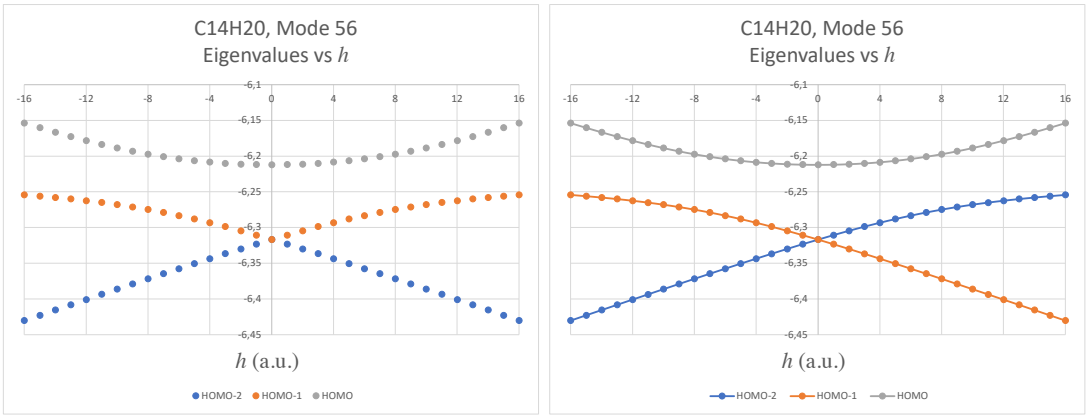

Figure 1: HOMO, HOMO-1 and HOMO-2 eigenvalues of diamantane as a function of the displacement parameter h, displaced through the normal mode 56 (1162 cm^{-1}). Left: As provided by the program `GenerateDataToPlotCrossovers.f90`; Right: After manual rearrangement of the data.

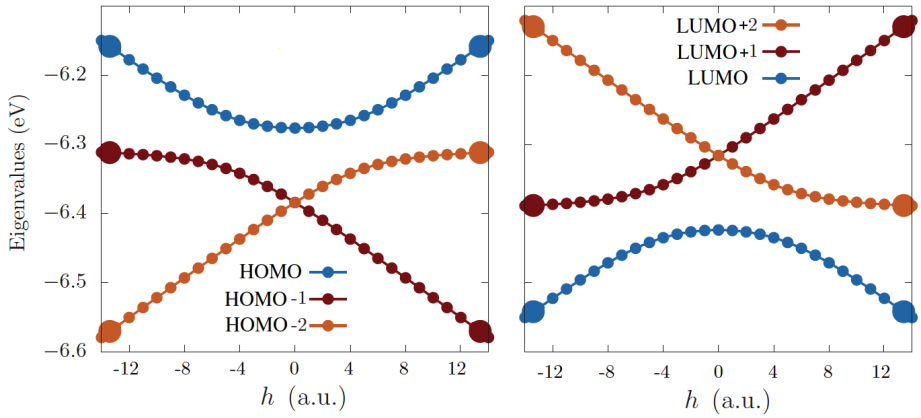

Figure 2: Qualitative shape of possible cases of anticrossing of three states.

`ToPlot/modeXX/stateYY±1.out`, `ToPlot/modeXX/stateYY±2.out`, eventually modified to make sure that the curves are smooth, see previous point). Then identify whether there is an anticrossing of 2 states or of 3 states (see Fig. 1 of the article). Then compile and run either `FPDoubleAnticrossing.f90` or `FPTripleAnticrossing.f90`.

The input file is `FPanticrossing.in`; an example of it is:

```
quantity_to_correct = HOMO   ! Either HOMO or LUMO
Nh = 33                      ! Total number of values of h
PolynomialDegreeg = 6        ! Degree of the fitting polynomial
Nintervals = 4097            ! number of intervals for the fitting
```

Required Input Files:

For an anticrossing of 2 states you must provide two input files called `eigenvalue1.dat` and `eigenvalue2.dat` containing the eigenvalue vs. h results.

For an anticrossing of 3 states, as shown in Fig. 2 the input file `eigenvalue2.dat` must contain the blue state. The other two are given in `eigenvalue1.dat` and `eigenvalue3.dat`.

Input Variables:

`quantity_to_correct`: Must be either `HOMO` or `LUMO` (regardless of the actual index of the analysed state). In case of anticrossing of three states, if the eigenvalues have a qualitative

shape like that of Fig. 2-left, then one must set quantity_to_correct = HOMO. Conversely, if the shape is that of Fig. 2-right, then one must set quantity_to_correct = LUMO.

`Nh`: number of considered values of the displacement parameter (33 in the case represented in Fig. 1, which corresponds to $h = -16, -15, ..., 0, 1, 2, 3..., 16$).

`PolynomialDegreeg`: degree of the polynomial used in the fitting of the eigenvalue-vs-h curves.

`Nintervals`: number of intervals (grid) used in the calculation of the couplings (the couplings are determined from the minimal distance between curves, and such distances are evaluated in as many points as `Nintervals`).

Output of the code

The output of this code contains the corrected eigenvalues (removing the unwanted effect of the anticrossing) for different values of h in the files: `eigenvalue1-bare.dat`, `eigenvalue2-bare.dat` and eventually (triple anticrossing) `eigenvalue3-bare.dat`.

- Fit the curve corresponding to your (bare $\simeq$ corrected) HOMO or LUMO[1] and extract the concavity *con* (second derivative, use e.g. xmgrace, Excel or OpenOffice).
- Calculate
$$con * (1000 * 219474.6305)/(2 * \omega_\nu) \quad (1)$$
with ω_ν in cm^{-1}. This is the correct contribution in meV of the analysed mode to the renormalization.
- In the file `out-FF-withcrossovers.out` (output of `frozen-phonon.x`) replace the value of the renormalization (last column in this file) due to the analysed mode with the corrected value.
- Sum up all the contributions of all vibrational modes given in the file `out-FF-withcrossovers.out` to obtain the total renormalization.

The authors are at the disposal of the community for inquiries and discussion. The software explained in this document is freely available for distribution and usage. We would appreciate citations to the original article [1] when used.

References

[1] P.G. Risueno, P. Han, S. Kumar and G. Bester, Physical Review B (submitted).

[2] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car,C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, et al., J.Phys.:Condens.Matter 21, 395502 (2009) http://dx.doi.org/10.1088/0953-8984/21/39/395502

[3] https://www.chemie.uni-hamburg.de/institute/pc/arbeitsgruppen/bester/downloads.html

[4] https://www.quantum-espresso.org/

[1] If you are calculating the correction (for anticrossing of 3 states) to the HOMO (or the LUMO), its eigenvalues-vs-h must be stored in `eigenvalue2.dat`, and the corrected values will be outputed to `eigenvalue2-bare.dat`. In other cases, e.g. if you want to calculate the correction to the HOMO-7 which participates in a triple anticrossing with HOMO-6 and HOMO-5, then the eigenvalue2 must be the eigenvalue (out of these 3) whose curve vs h is similar to a parabola –see Fig. 2–, i.e. it has no straight stretch. If this nearly-parabolic state is the highest one of the three, set quantity_to_correct = HOMO in `FPanticrossing.in`; if it is the lowest one, set quantity_to_correct = LUMO.