A tutorial for electron-phonon renormalization calculations without errors due to crossing and anticrossing of quantum states

Pablo García-Risueño[1,*]

[1] *Institut für Physikalische Chemie, Universität Hamburg,*
Grindelallee 117, D-20146 Hamburg, Germany //
Universidad de Zaragoza, C/ Mariano Esquillor s/n, 50018, Zaragoza, Spain

[*] E-mail:`risueno@unizar.es`,`garcia.risueno@gmail.com`

The frozen-phonon method [1, 2] makes it possible to calculate the renormalization of electronic eigenvalues due to the nuclear vibrations. In the present tutorial we explain in detail how to perform the calculations necessary to find the frozen-phonon renormalization of electronic eigenvalues of molecular (non-bulk) systems. The calculations use both `Quantum Espresso` (QE) [3] executables (`pw.x`, `ph.x`, `pp.x`, `dynmat.x`) and also programs written by Pablo Garcia Risueño (`vib_potPGR.x`, `frozen-phonon.x`, `FPDoubleAnticrossing.f90`, `FPTripleAnticrossing.f90`). Ref. [2] describes a widely used frozen-phonon method for the calculation of electronic eigenvalues renormalized due to the interaction of the electrons with vibrating nuclei. However, the method presented in [2] makes errors due to the mixing of states; in [1] we present a refinement of the method of [2] to avoid these errors. The methods and programs presented in this tutorial perform calculations following the method of [1]. The explanations of the calculation methods presented in this document, together with the programs downloadable at [4], are expected to present clear and detailed indications so that accurate calculations frozen-phonon renormalizations can be easily done, with minimal (human) effort.

The frozen-phonon calculations consist of several stages:

1. Geometry optimization (i.e. relaxation).

2. Calculation of the phonon frequencies and vibrational eigenvectors.

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

4. Calculation of undisplaced and displaced electronic eigenvalues and wavefunctions.

5. Calculation of renormalization of electronic eigenvalues due to nuclear vibrations (i.e. frozenphonon calculation itself).

6. If necessary, correction of the anticrossing effects as explained in Ref. [1].

Below we explain the stages enumerated above for a test system (diamantane, $C_{14}H_{20}$), and we provide the scripts and programs to run the corresponding calculations (they require an installation of `Quantum Espresso`). If calculations are performed correctly, the method

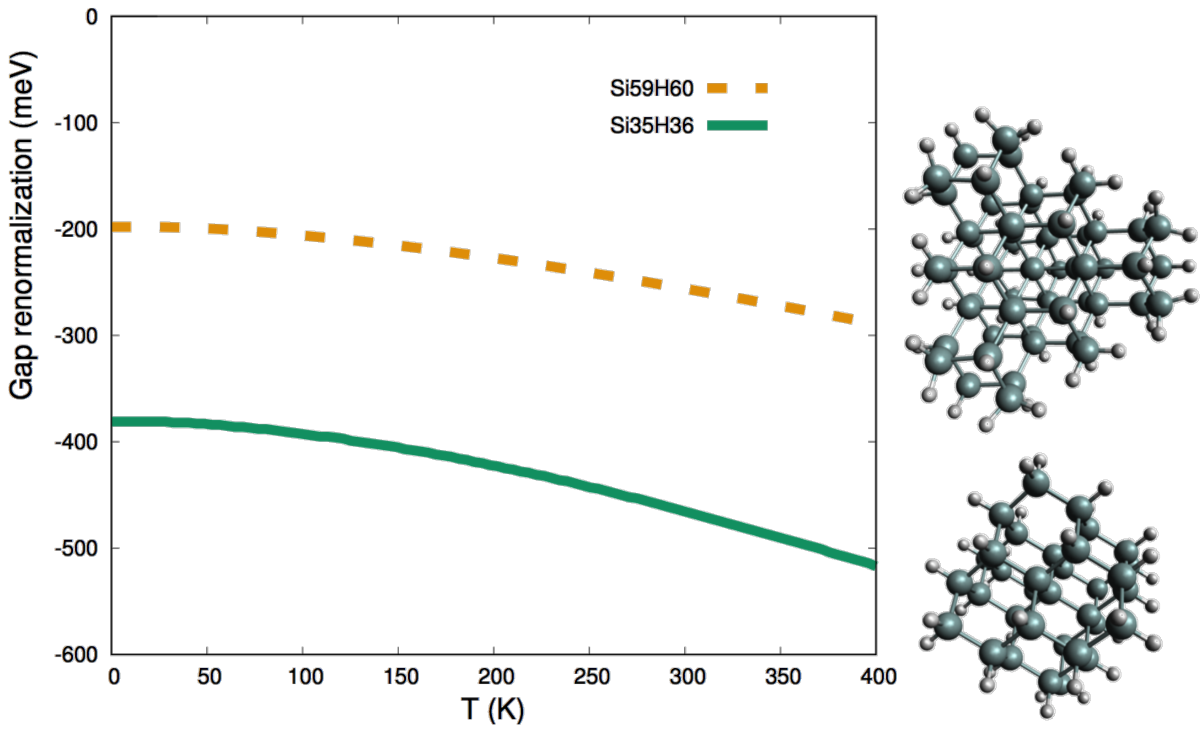

FIG. 1: Example of gap renormalizations as a function of the temperature for two molecules.

explained in this tutorial provides the zero-point renormalization of gaps of molecular systems, as well as their temperature-dependent expressions, like those presented in fig. 1 below.

The input files, scripts and `Fortran` programs necessary to perform the calculations of this tutorial, as well as the files with the atomic coordinates of the analysed molecules (`.xyz` files), can be downloaded from [4]. Download them all into a parent directory where you will run the calculations of this tutorial, and copy all the files stored in the scripts folder into this parent directory. Remember to do `$ chmod 777 *.sh` (at the terminal shell) to give the scripts permission to work. We recommend that you also copy the files of the folder called pseudopotentials that you just downloaded into the directory which stores the pseudopotentials from your `Quantum Espresso` installation. Reference outputs obtained with these inputs can be downloaded from [4] (note that they are just a reference, your output does not need to be identical to this one because of light differences due to different versions of the program and to some random component in the executions of `Quantum Espresso`). The example scripts of this tutorial are devised to be run in a given cluster (whose name is HUMMEL); an example script to be run in another kind of cluster lies in `example_run_in_XBes.sh`.

I (PGR) tried to make this tutorial as clear as possible; if you have any doubt, please don't hesitate to contact me at any of my email addresses (see above).

IMPORTANT NOTICE: The directories where the executables and pseudopotentials of Quantum Espresso of this tutorial are stored will be different from your own directories. This means that:

- In all your task submission scripts (.sh files) you will have to write `export ESPRESSOLOCATION=directory_where_your_pw.x_file_lies` instead of `export ESPRESSOLOCATION=/home/fcnv733/PROGRAMS/espressoHahn/bin/`
- In several of the `Quantum Espresso` input files (`geomopt.in`, `scf.in`) it is necessary to specify the directory where the pseudopotential files (.UPF files) are stored. In these files, replace `pseudo_dir='/home/fcnv733/PROGRAMS/espressoHahn/pseudopotentials/'`, with `pseudo_dir='directory_where_your_pseudopotentials_are_stored'`,

1. GEOMETRY OPTIMIZATION (RELAXATION)

First of all we must find the values of the nuclear (ionic) coordinates which correspond to a minimum of energy and make forces on nuclei (ions) low. This is an essential condition to find the phonon frequencies (step 2.), because the phonon theory (dynamical equation) relies in the harmonic approximation (the energy as a function of the nuclear positions is expanded in a Taylor series of degree 2). Prior to the step of geometry optimization we recommend to perform convergence calculations in the plane-wave cutoff and box size parameters (`ecutwfc` and `a` from `Quantum Espresso`). Note, however, that a cutoff of 30 Ry is enough for the systems that we have tested (e.g. in diamondoids it is necessary to reach 110 Ry to converge the HOMO-LUMO gap from DFT calculations with QE, but a cutoff of 30 Ry suffices for converging the frozen-phonon renormalization). The box size should be high enough to avoid the undesired effect of periodic images; usually it suffices that there is a distance of about 6 Angstrom between the most external atoms of the molecule and the box wall (in this tutorial we take a much larger box, with `a = 35,` but this is not necessary). You can reduce the size of the box to save computing time. ADVICE: Note that our program to perform frozen-phonon calculations demands cubic boxes, other box shapes are forbidden. Hence make sure that in the `&system` field of your `Quantum Espresso` input files you write `ibrav = 0,` (this specifies that the box is cubic). You must also pay attention to the field for the number of bands, `nbnd` ; it must be at least 20% higher as half the number of electrons of the system (we recommend that there are at least 20 unoccupied states). For example, in $C_{14}H_{20}$ we have 76 electrons and hence 38 bands, and we take `nbnd = 51,` .

The geometry optimization itself can be done running the `run_go.sh` script with `Quantum Espresso`. To do it, just type in the terminal: `$ sbatch run_go.sh` (or `$ qsub example_run_in_XBes.sh` in other clusters).

The output that you obtain in this relaxation calculation (`out_geomopt.out`) should look similar to the one available in the reference-outputs folder that you downloaded. It is very important to make a good relaxation, making the total final force low. All the individual x-, y-, z-components of the force on each atom should have modules not being much higher than 0.00001000 Ry/au in the `Quantum Espresso` output (`outputfiles/out_geomopt.out`); values below `Total force = 0.000120` in the output file from `Quantum Espresso` are typically accurate enough, but it is better to make this total force as small as possible (note that this may require a high number of iterations and a non-negligible execution time). To this end, one may need to repeat the process of relaxation several times. You can do it by copying the final coordinates block of the output (`outputfiles/out_geomopt.out`) of the i-th run into the input file (`inputfiles/geomopt.in`) for the (i+1)-th run. To reduce the `forc_conv_thr, etot_convthr` and `conv_thr` (never beyond e-17) may also be necessary. If the run fails, try to replace the `diagonalization = 'david'` eigensolver by `diagonalization = 'cg'`. If the relaxation is hard to converge, make `mixing_beta` lower, till 0.1. Note that nothing guarantees that an appropriate relaxation of your system will be obtained; for some systems it is impossible to get it with some functionals; for other systems it may be impossible at all.

If you would like to understand well the meaning of the variables appearing in the QE input files, visit `http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PW.html` . If you want to use other pseudopotentials (e.g. to perform GGA instead of LDA calculations), you can download them from `http://www.quantum-espresso.org/pseudopotentials/` (you will also need to copy the .UPF files to the directory where your Quantum Espresso installation stores them, and to specify the name of the new .UPF file in the input files of QE, e.g. replacing C 12.0107 `C.pz-vbc.UPF` by C 12.0107 `C.pbe-hgh.UPF`).

2. CALCULATION OF THE PHONON FREQUENCIES AND VIBRATIONAL EIGENVECTORS

Once an appropriate relaxation has been performed, we can solve the so-called dynamical equation [5] to find the phonon frequencies and vibrational eigenmodes (the vectors which indicate the vibration directions of the nuclei). To this end, we need to run 3 different executables from `Quantum Espresso` using 3 different input files.

A. Ground state calculation

First,we need to perform a ground state calculation (the input file is `scf.in`, the used executable is `pw.x`). Note that you have to copy the final coordinates of out_geomopt.out into the `ATOMIC_POSITIONS { angstrom }` block of `scf.in`. The nuclear positions must be in Angstrom; make sure that one reads:

```
ATOMIC_POSITIONS { angstrom }
```

and not

```
ATOMICPOSITIONS ( angstrom )
```

Note that we have changed two fields of `&electrons` in `scf.in` with respect to `geomopt.in`: we have made `convthr = 1.0e-12,` (to save computing time, without dip of accuracy) and we have made `startingwfc='atomic',` (this line is very important, because by default Quantum Espresso includes random components in the starting guess to solve the Kohn- Sham equations, and this could lead to inconsistencies in the results necessary for the frozenphonon calculations, which are obtained in different runs and they could hence have different offsets in the eigenvalues).

B. Solution of the dynamical equation

Then we need to solve the dynamical equation (input file dyneq.in, with the ph.x executable). Usually a tolerance of tr2_ph=1.0d-14, in dyneq.in is accurate enough. If the calculation was not finished after the maximum execution time (e.g. 12 hours), continue with it (just ph.x, not pw.x) just by adding recover=.true. in dyneq.in

C. Modification of the results to satisfy summation rules

Finally we can modify the results to make them satisfy certain sum rules (the input file is dynmat.in, the used executable is dynmat.x). One can make the three runs stated above (2.1, 2.2, 2.3) with a single script like run_dyneq.sh If the calculations were properly run, you will find output files containing phonon frequencies and vibrational eigenmodes (dynG, dynmat.out) in the directory where you are running the calculations. You can then check that the calculations were properly run. First of all, check that the ground state calculation used to perform your phonon calculation is correct; to this end, read the outputfiles/out_scf.out file; look for the line where one reads Total force, and make sure that there is only one line with these words, and that its value is lower than 0.000120. If that is not the case, you will have to repeat the geometry optimization until you reach an appropriate relaxation. Then you have to check that the obtained phonon frequencies are reasonable. Open the dynG file and look for the freq (7) text; this is the 7th lowest phonon frequency, which is the lowest one with a physical meaning (the lowest 6 frequencies correspond to translations and rotations of the system, and they are unphysical). Make sure that this 7th frequency is above 150 cm^{-1}. Repeat the same procedure with dynmat.out. If frequencies above the 6th one are negative, this means that they are unphysical, and that the result is not valid. You can try to avoid the error by performing a better relaxation or by solving the dynamical equation more accurately, using `tr2_ph` in `dyneq.in` which is lower than 1.0d-14 (some systems demand values as low as 1.0d-20; for the system of this tutorial, 1.0d-17 is appropriate). Usually the results from `dynG` and from dynmat.out will be pretty similar (and both physically meaningful); if that is the case, we will take `dynmat.out` to generate the displaced nuclear positions. If, on the other hand, the 7th frequency of `dynmat.out` is below 150 cm^{-1} but that of `dynG` is not, we use `dynG` instead (writing `FreqsUfile = dynG` in `vib_potPGR.in`).

3. GENERATION OF FILES WITH DISPLACED NUCLEAR POSITIONS FOR THE FINITE-DIFFERENCE (I.E. FROZEN-PHONON) CALCULATIONS.

Now is the time to generate the input files with displaced nuclear positions which are necessary to find the electronic eigenvalues at the displaced positions and, in turn, to perform the frozenphonon (finite-difference) calculations. To this end we will use a code written by us which is called `vib_potPGR.f90`. This program makes copies of `scf.in` (where the relaxed positions appear) but with nuclear positions displaced through the normal modes (i.e. the vectors which give the vibration of the nuclei, which are the U eigenvectors of the dynamical matrix, see [5]). These files are stored in two folders, called `frozen+R`and `frozen-R`(for positive and negative displacements). If X_0 are the relaxed positions of the nuclei, the files in `frozen+R`and frozen- R contain positions being $X_0 \pm h \cdot U_\nu$, being ν the index of normal mode (running between the unity and 3 times the number of atoms) and being h one scalar whose value is determined by the user; we call h the finite-difference parameter, and it can be specified in the `Finite_Difference_Parameter` field of `vib_potPGR.in`. In [1] we recommend to use a value of 4.0000 for it, but the user can freely decide what is the value which he or she chooses (4.0000 is expected to provide maximum accuracy to the results). WARNING: Make sure that the `Finite_Difference_Parameter` (h) of `vib_potPGR.in` is the same as that of `frozenphonon.in`, for otherwise the calculated renormalizations would be wrong.

To generate the `Quantum Espresso` input files with displaced positions, proceed as follows. In the terminal, in the directory where you are running the calculations, copy:

```
$ cp programs/* .
$ cp inputfiles/Veff-unperturb.in .
$ cp outputfiles/out_scf.out .
$ cp inputfiles/scf.in .
```

Make sure that `startingwfc='atomic',` is written in `scf.in`, and write this text if it lacks. Now compile the code necessary to generate the input files (`vib_potPGR.x`) and run it and run by doing as indicated below (specifying the compiler of your computer:

```
$ module switch env env/intel-15.0.3_impi-5.0.3
$ ifort -o vib_potPGR.x vib_potPGR.f90
$ ./vib_potPGR.x
```

An appropriate run will produce an output like the one of `output-reference/out_vibpotPGR.out`, as well as a file called `Uvec.dat`, which contains phonon frequencies, which will be necessary later (step 5.). The program `vib_potPGR.x` requires an input called `vib_potPGR.in`. For systems other than the one presented in this tutorial, make sure that you modify `vib_potPGR.in` to include the correct number of atoms of each species and their masses (in a.m.u.'s) in the previously stated, you can use either `dynG` or `dynmat.out` for doing the displacements (field `FreqsUfile` of `vib_potPGR.in`).

4. CALCULATION OF UNDISPLACED AND DISPLACED ELECTRONIC EIGENVALUES AND WAVEFUNCTIONS

As previously stated, the frozen-phonon method calculates the renormalization of electronic eigenvalues due to electron-vibrational interaction by performing finite-difference displacements of the ions along the vibrational normal modes (displacements of $\pm h \cdot U_\nu$, being U_ν the eigenvectors of the dynamical matrix and being h an adimensional finite-difference parameter to be fixed). The more accurate method for frozen-phonon calculations that we present in [1], in contrast to that of [2], demands the evaluation of electronic wavefunctions and of their overlaps; low overlaps between the undisplaced and displaced wavefunctions indicate that the calculated eigenvalues are reliable; if, on the other hand, overlaps are relatively low, then the calculation will not be reliable, and we will have either to correct it by hand or to discard it (for the given mode which provides low overlap).

We need thus to perform DFT runs where both the electronic eigenvalues and the electronic wavefunctions are calculated and stored. To this, make sure that you use the same prefix in `wfcs.in` than in `scf.in` (diam- in this tutorial). To reduce the wall-clock-time of the execution of this step, we are going to perform its calculations using 5 scripts which will run in parallel: `run-wfcs0.sh`, `run-wfcs-a.sh`, `run-wfcs-b.sh`, `run-wfcs+a.sh` and `run-wfcs+b.sh`. The first script (`run-wfcs0.sh`) calculates eigenvalues and wavefunctions for the relaxed nuclear positions X_0; the rest, for displaced nuclear positions. In these scripts, one can find blocks similar to the one presented below:

```
  ### Variables and paths (change them to the value appropriate for your
system) ########
  export LD_LIBRARY_PATH=/home/fcnv733/PROGRAMS/libmkl_PGR/
  module switch env env/intel-15.0.3_impi-5.0.3
  export ESPRESSOLOCATION=/home/fcnv733/PROGRAMS/espressoHahn/bin/
  export PREFIX=diam-
  InitialBandFF=36 # Initial electronic orbital whose overlap with HOMO
and-or LUMO
  is to be checked
  FinalBandFF=39 # Final electronic orbital whose overlap with HOMO and-or
LUMO is to be checked
  initial_mode=7 # First phonon mode of this run
  final_mode=50 # Last phonon mode of this run
  ########################################################
```

Let us briefly explain the meaning of the variables of the block above. The first one (`LD_LIBRARY_PATH`) specifies the path to the `mkl` library (in clusters like the one employed for the calculations of this tutorial, `HUMMEL`); this library is necessary for `Quantum Espresso` to run properly. The line below is also HUMMEL-characteristic and it activates the intel

Fortran compiler. The line below, as previously explained, determines the location of the folder which stores the executables of Quantum Espresso. The PREFIX variable must be equal to the prefix field which appears in the input files of QE (scf.in, geomopt.in and wfcs.in,); IMPORTANT: it is essential that these two variables have identical value, for otherwise the sought output displaced eigenvalue files would NOT be properly stored. The variables called InitialBandFF and FinalBandFF are the indices of electronic orbital that we will consider in the comparison of overlaps between undisplaced and displaced wavefunctions. Their values must be chosen as a function of the number of electrons of the system and of the distances of HOMO and LUMO with the neighboring eigenvalues. For example, in this case our system is $C_{14}H_{20}$, and hence it has 76 electrons; since we do not consider spin, the HOMO will be the orbital number 76/2=38, and the LUMO will be the orbital number 39. If we check the eigenvalues in out_scf.out, we will see that neither the HOMO nor the LUMO are degenerate. Hence, in a first approximation, we could fix InitialBandFF=38 and FinalBandFF=39. The electronic eigenvalues in a QE output file like out_scf0.outappear immediately below of the text bands (ev). In the case of this tutorial, we see a very large distance (of nearly 400 meV) between the LUMO and the LUMO+1. Hence we can guess that the overlap between LUMO and LUMO+1 is going to be very low, and we can keep FinalBandFF=39. The difference between HOMO and HOMO-1 is much lower (about 100 meV); hence, we decide to print the wavefunctions of HOMO-1 and HOMO-2 to calculate their overlaps with the HOMO (just to know them, this is a fully personal decision), and we write InitialBandFF=36. Finally, we distribute the calculation of different modes for positive and negative displacements into different execution scripts; the initial_mode and final_mode variables determine which range of vibrational modes are calculated in a given shell script. Note that one must calculate all physically meaningful modes, this is, between 7 and (3 times the number of atoms of the system), which is 102 in the case of $C_{14}H_{20}$.

If the calculations were properly run, the following files should appear (they should be similar to the files with the same name that you can find in the output-reference folder):

- a file called eigenval.xml ; it contains the eigenvalues of the relaxed system in atomic units with many digits; another file called out_scf0.out, with data about the calculation, should also be there.

- two folders called outputfiles/outputsVeffcalc/displaced- and outputfiles/outputsVeffcalc/displaced+, each containing 96 files called `eigenval-modeXXX.xml` and other 96 files `out_scfmodeXXX.out` (with XXX running between 7 and 102; 96 is the number of physically meaningful phonon frequencies of this system, 96 = (14+20)*3 - 6).
- 4 wavefunction files called `wfc36.wfn`, `wfc37.wfn`, `wfc38.wfn`, `wfc39.wfn` in the `Wfcsundisplaced/` folder; they contain the wavefunctions for undisplaced (relaxed) nuclear positions X_0 of LUMO, HOMO, HOMO-1 and HOMO-2 (because we specified to print orbitals between 36 and 39).
- 384 wavefunction files in the `Wfcs/Wfcs-displaced-` folder, and the same in `Wfcs/Wfcsdisplaced+`. They correspond the the wavefunctions calculated at displaced positions for different vibrational modes: $X_0 \pm h \cdot U_\nu$, being ν the index of normal mode.

5. CALCULATION OF RENORMALIZATION OF ELECTRONIC EIGENVALUES DUE TO NUCLEAR VIBRATIONS

We have reached the final step (if no corrections due to anticrossing are needed), where we calculate the renormalization of electronic eigenvalues (HOMO and LUMO) due to electron-vibrational interaction. To perform this calculation, you will need the files listed at the end of the previous section lying in the directory where you run the calculations. In addition, you will need the code of the program to do the calculation (frozenphonon. f90), the input file for it (`frozenphonon.in`) and the`run-FF-withcrossovers.sh` , `run-FF-wocrossovers-HOMO.sh` , `run-FF-wocrossovers-LUMO.sh` scripts. The input file for this program is called `frozenphonon.in`; make sure that you specify the same value for h -the finite-difference parameter, field `Finite_Difference_Parameter` - that you specified in `vib_potPGR.in`. The meaning of all the fields of `frozenphonon.in` is explained in an appendix below. You may need to enable the `Fortran` compiler e.g. with:

```
$ module switch env env/intel-15.0.3_impi-5.0.3
```

We then perform a first fast calculation of the frozen-phonon renormalization, not calculating overlaps of wavefunctions. We make `Check_crossover = 0` in `frozenphonon.in`. Then we do:

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

The script above compiles the code (`frozen-phonon.f90`)) and runs it. A few seconds later, a file called `out-FF-withcrossovers.out` will appear. Its last 12 lines should provide similar values to those of the file with the same name in outputfiles-reference, i.e. a renormalization of the HOMO of about +433 meV and a renormalization of the LUMO of about -113 meV. At the end of this output file you should look something like what follows (do e.g. $ tail -111 out-FFwithcrossovers. out):

```
============= HOMO =====================
original: -2.66753 eV - renormalized: -2.17441 eV
difference: 432.8 meV <=====
========================================

============= LUMO =====================
original: 2.66753 eV - renormalized: 2.55996 eV
difference: -113.5 meV <=====
========================================
```

The gaps provided by DFT calculations (i.e. the values of HOMO and LUMO) are not very accurate; the physically meaningful quantities are the frozen-phonon renormalizations, which are the numbers written after the word ”difference”. This is the goal of our calculation: we have found the zero-point renormalization (because we imposed T=0) of the band gap of diamantane ($C_{14}H_{20}$), which would be obtained by subtracting both differences. Then we can perform a 2^{nd} calculation of the frozen-phonon renormalization which evaluates the overlaps between undisplaced and displaced wavefunctions (i.e. at X_0

and at $X_0 \pm h \cdot U_\nu$). The details of the interpretation of overlaps are presented in [1]. To this end, make `Check_crossover = 1` in `frozenphonon.in`. To reduce the wall-clock time, we can run in parallel the calculations of the HOMO and of the LUMO. First we state the bands of the HOMO and below that we have printed (36 to 38 in our case), and we make `InitialStateFPh = 36` and `FinalStateFPh = 38` in `frozenphonon.in`. Then we submit:

```
$ sbatch run-FF-wocrossovers-HOMO.sh
```

When we are sure that this is already running, we can modify `frozenphonon.in` file to calculate the LUMO; we establish the band limits to go from the LUMO to the highest printed orbital (39 to 39 in our case) doing `InitialStateFPh = 39` and `FinalStateFPh = 39` in `frozenphonon.in`, and we submit:

```
$ sbatch run-FF-wocrossovers-LUMO.sh
```

If everything went properly, you will be able to read the files with the overlaps of wavefunctions while they are being produced; their names are `wfc_product_results-HOMO.out` and `wfc_product_results-LUMO.out`. As explained in [1], for a finite-difference parameter h equal to 2 if the squared overlap between the undisplaced and displaced (for a given mode) HOMO is > 0.995, then we can consider the frozen-phonon calculation due to this mode to be reliable (this corresponds approximately to values in wfc_product_results-HOMO.out above 0.9975). For many systems this condition does always hold, and hence we can trust the result of the frozen-phonon calculation without further processing. For other systems, however, there are a few modes where the squared overlap between undisplaced and displaced HOMO (or LUMO) is lower than 0.995 (for h=2). The corrections obtained for these modes are not fully reliable; to find reliable contributions one must perform calculations of the eigenvalues for displacements through these problematic normal modes for different values of h, as explained in [1]. In the example of this tutorial, this happens for 6 of the modes (which may be the modes 53, 54, 55, 56, 61 and 62), which is quite uncommon. Finally, one can calculate the frozen-phonon renormalizations of HOMO and LUMO as a function of the temperature by uncommenting the variables called Initial_Temperature, Final_Temperature and Number_of_Temperatures in `frozenphonon.in` and giving them desired values (with value of Final_Temperature lower to the melting point of the analysed material). The data corresponding to the renormalizations for different temperatures will appear at the output file. If one wants to calculate the renormalization of other states different from HOMO and LUMO, one can do it using a trick consisting in artificially changing the degeneracy of the HOMO or the LUMO using the HOMO_imposed_degeneracy , LUMO_imposed_degeneracy variables in `frozenphonon.in` (see an explanation of them at the end of the appendix below). To find the frozen-phonon renormalization of the LUMO+n one makes LUMO_imposed_degeneracy = n+1 and then one multiplies the result shown at the screen by (n+1). For example, we know that the LUMO (39th state) of diamantante ($C_{14}H_{20}$) is NOT degenerate; however, the HOMO-LUMO transition is forbidden, and hence we would like to know the renormalization of the LUMO+1. We can calculate it by making LUMO_imposed_degeneracy = 2 . When we run `frozen-phonon.x` we obtain:

```
The contribution of state 39 is -56.7648 meV
The contribution of state 40 is -46.5749 meV

======== LUMO =======================
original: 2.90883 eV - renormalized: 2.80550 eV
difference: -103.3 meV <=====
====================================
```

The renormalization of the LUMO+1 (40th state) is twice -46.57, i.e. -93.14 meV (we have to multiply by the imposed degeneracy because when a state is degenerate, its frozen-phonon renormalization is indeed the average of the degenerate states). Note that in this case, where the degeneracy is a fabrication, the presented result for the LUMO (the line difference: -103.3 meV <===== above) is no longer physically correct. To find the renormalization of the HOMO-m we just make HOMO_imposed_degeneracy = 1+m and we multiply the obtained result by (1+m).

6. CORRECTION OF DISTORTIONS DUE TO ANTICROSSING EFFECTS

A. Introduction

Some molecules present undesired anticrossing effects which can distort the renormalizations calculated in section 5 so much that they become totally wrong. In Reference [1] we explain in detail the problem and the manner to correct it. It is desirable to read and understand Ref. [1] if one wants to perform frozen-phonon calculations. In this section we present a practical example on how to perform such calculations; this section is a complement to the contents of Ref. [1].

B. Checking if there is distortion due to anticrossings

First of all, we have to check whether it is likely that the molecule that we have analysed ($C_{14}H_{20}$ in the example of this tutorial) presents distortion due to anticrossing. We can notice it using two methods:

i) Outliers in the renormalization values: In the output document of `frozen-phonon.x`, one can find `Ndeg` times (3N-6) rows –where `Ndeg` is the degeneracy of the analysed level– for the HOMO and for the LUMO, where the last column of the ν-th row indicates the contribution of the ν-th mode to the renormalization of the anlysed level. Typical values for the contribution of each mode is of the order of meV. If a few of the modes (e.g. six) display a contribution much higher than the rest (say of the order of tens of meV) it is likely that they correspond to a mode distorted by anticrossing effects. In the example that we analyse of $C_{14}H_{20}$, the eight modes which are distorted by anticrossing have contributions (uncorrected, i.e. those obtained after step 5) of between 20 and 53 meV, much higher than the average (which is about 6 meV; this average is itself too high due to the distortions; usually the contribution of each mode should be below 2 meV).

ii) Low overlapping of wavefunctions: In the output file of the calculations made with << `Check_crossover = 1` >> in the `frozenphonon.in` file you can check how high the overlapping between the wavefunction calculated with undisplaced positions and the wavefunction calculated with positions displaced in the direction given by the ν-th normal mode. If you compare the overlaps of the different modes and notice that for some of them the overlapping is clearly lower than for the rest (e.g. 0.94 vs 0.999), then this is a hint that there may be an anticrossing effect.

If you have found any of the effects mentioned above, you should make sure that there is no problem with anticrossing effects. This will take a bit computer and human time. As an example, let us consider that we found six potentially problematic modes in $C_{14}H_{20}$ (with LDA-PZ functional, vbc pseudopotential, 30 Ry of plane-wave cutoff), which are the modes 53, 54, 55, 56, 61 and 62. For these modes, we will have to recalculate the eigenvalues for displaced positions for a set of displacement sizes (e.g. h=1, 2, 3, 4, 5, 6, 8, 10, 16 or h=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16). It is important to take at least one value equal to or above 16 because it will be later necessary to find oblique asymptotes (as those displayed as dashed lines in Fig. 1-e of [1]). Please, rerun `vib_potPGR.x` with the corresponding h values in its input file (`vib_potPGR.in`). You must do this sequentially; this is: create directories called DP_01, DP_02, DP_03, etc., and then copy the corresponding files, modify `vib_potPGR.in` in each of them (this is, write `DisplacementParameter = 1.000000000`, `DisplacementParameter = 2.000000000`, etc. in it), and run the calculations corresponding to the problematic modes (do not analyse the other modes, that would be a waste of computing time). Note that, as you did in the previous sections, you must perform the calculations with both positive and negative directions of the displacement.

If we have noticed that there may be distortion due to anticrossings, we need to perform the calculations presented in the sections 6 C and 6 D below (otherwise they can be omitted).

C. Preparation of the data

In this section we explain how to prepare the data for the calculation of the corrections due to anticrossings. You need a directory which contains the files called `Uvec.dat` (with the normal modes and frequencies), `eigenval.xml` (with the eigenvalues of the ground state calculation with `Quantum Espresso`) and `out_scf.out` (with the output of the ground state calculation with Quantum Espresso). Such directory must also include directories (DP01,DP02,...,DP10,..., where ”DP” stands for *displacement parameter*) which include the eigenvalues for the displaced positions. Each of these subdirectories must contain two further subdirectories called displaced+ and displaced-. In the displaced+ and displaced- folders one must find as many .xml files as vibrational modes to analyse there are. This is, these folders must contain eigenval-mode7.xml, eigenval-mode8.xml, ... (in the example that we analyse, diamantane, we analyse till eigenval-mode102.xml). These files are obtained running simulations as those explained above.

When the referred files are ready, please copy the files GenerateDataToPlotCrossovers.f90 and GenerateDataToPlotCrossovers.in to the directory where they lie. An example of the latter file –corresponding to the example of diamantane– is:

```
Number_of_atoms = 34
Number_of_species = 2
Read_from_xml_file = 1
elec_eigval_file_name = out_scf.out
Band_occupation = 2
Modes_to_analyse######
53
54
55
56
61
62
##################

Displacement_parameters######
01
02
03
```

```
04
05
06
07
08
09
10
11
12
13
14
15
16
##################
```

The first variable (`Number_of_atoms`) indicates the number of atoms of the system (in this case 34, 20 hydrogens and 14 carbons). The second variable (`Number_of_species`) indicates the number of different atomic species (2 in this case, because there are C and H atoms; for urotropine it would be 3, because there are C, N and H, for fullerenes it would be 1, because they consist of just C, etc.). The third variable (`Read_from_xml_file`) indicates that the data must be read from `.xml` files, instead of the output of `Quantum Espresso` in text file (please, set this variable to 1 always; setting it to 0 leads to severe inaccuracies). The fourth variable indicates the name of the output of `Quantum Espresso` of the ground state at the relaxed positions (which contains some necessary information). The fifth variable indicates that each eigenvalue has an occupancy of 2 (we do not split by spin). The next two blocks indicate the vibrational modes to analyse (those which we suspect present problems of anticrossing) as well the values of the displacement parameters which were used in the calculations (please, do not forget to write a "0" first if $h < 10$).

When the files are gathered and the input file is defined as indicated above, compile and run the `GenerateDataToPlotCrossovers.f90` program, e.g. making in a terminal:

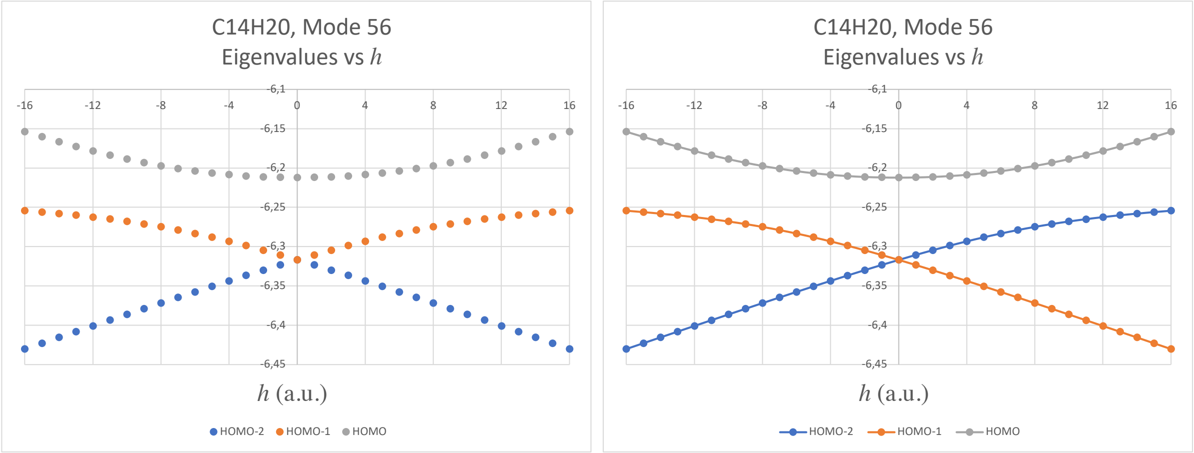

FIG. 2: HOMO, HOMO-1 and HOMO-2 eigenvalues (i.e. eigenvalues 38, 37 and 36) of diamantane (from LDA-PZ functional, cutoff 30 Ry) as a function of the displacement parameter h, displaced through the normal mode 56 (1162 cm^{-1}). Left: As provided by the program `GenerateDataToPlotCrossovers.f90`; Right: After manual shift (exchanging the values for positive h of eigenvalues 36 and 37 so that curves are smooth).

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

If after doing so you receive no error message, you can go to the folder called `ToPlot` and select one of the folders in it (e.g. `ToPlot/mode56`). In it you will find files of the different eigenvalues vs h. Select those which involve the eigenvalues which you suspect are distorted by anticrossings; in this example the HOMO (38th state), HOMO-1 (37th state) and HOMO-2 (36th state). Select the data of all three and write them to a table like Tab. I.

The data displayed on Tab. I are plotted in Fig. 2-left. Note that it may be necessary to exchange the values of HOMO-2 and HOMO-1 for some values of h to make the curves smooth (Fig. 2-right). In addition, note that this example corresponds to an anticrossing of 3 states; in other cases the anticrossing involves just two states, and hence one must just select the data of two states (e.g. HOMO and HOMO-1).

h	Eigv.36 (HOMO-2)	Eigv.37 (HOMO-1)	Eigv.38 (HOMO)
-16	-6.430498025	-6.254399585	-6.153912529
-15	-6.4231426	-6.256168043	-6.160406631
-14	-6.415793552	-6.258094162	-6.166678797
-13	-6.40845435	-6.260211297	-6.172690795
-12	-6.401128748	-6.262556968	-6.178401177
-11	-6.393822139	-6.265171832	-6.183766085
-10	-6.386539463	-6.2680949	-6.188739865
-9	-6.379286659	-6.271362776	-6.193281394
-8	-6.372070707	-6.275002831	-6.197354539
-7	-6.364899931	-6.27903268	-6.200934406
-6	-6.357784497	-6.283456058	-6.204007754
-5	-6.350734742	-6.288261937	-6.206573541
-4	-6.343764929	-6.293428443	-6.208640367
-3	-6.336890261	-6.298925489	-6.210222402
-2	-6.330129825	-6.304717467	-6.21133788
-1	-6.323505329	-6.310767603	-6.212000524
0	-6.317127656	-6.316975203	-6.212202682
1	-6.310767603	-6.323505329	-6.212000524
2	-6.304717467	-6.330129825	-6.21133788
3	-6.298925489	-6.336890261	-6.210222402
4	-6.293428443	-6.343764929	-6.208640367
5	-6.288261937	-6.350734742	-6.206573541
6	-6.283456079	-6.357784505	-6.204007765
7	-6.27903268	-6.364899931	-6.200934406
8	-6.275002831	-6.372070707	-6.197354539
9	-6.271362776	-6.379286659	-6.193281394
10	-6.2680949	-6.386539463	-6.188739865
11	-6.265171832	-6.393822139	-6.183766085
12	-6.262556968	-6.401128748	-6.178401177
13	-6.260211297	-6.40845435	-6.172690795
14	-6.258094162	-6.415793552	-6.166678797
15	-6.256168043	-6.4231426	-6.160406631
16	-6.254399585	-6.430498025	-6.153912529

TABLE I: Example of eigenvalues vs displacement parameter (diamantane, LDA-PZ, displacement through normal mode 56, whose frequency is 1162 cm^{-1}).

D. Actual calculation of the correction

1. Introduction

After collecting the data of eigenvalues as a function of h, we must make a plot of them (such as Fig. 2-right) in order to make a visual inspection to check whether the anticrossing

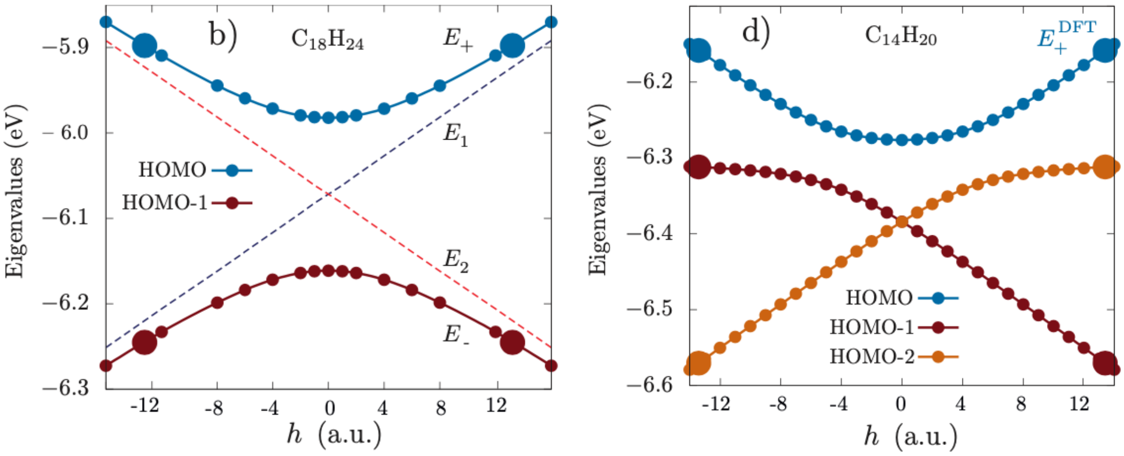

FIG. 3: HOMO, HOMO-1 and HOMO-2 eigenvalues (i.e. eigenvalues 38, 37 and 36) of diamantane (from B3LYP functional) as a function of the displacement parameter h, displaced through the normal mode 62. Left: As provided by the program `GenerateDataToPlotCrossovers.f90`; Right: After manual shift (exchanging the values for positive h of eigenvalues 36 and 37 so that curves are smooth).

involves 2 or 3 states. In both cases, the curves of eigenvalue-vs-h differ from parabolas. However, their shapes are very different; the patterns of anticrossings of 2 and 3 states are displayed in Fig. 1-b and Fig.1-d of Ref. [1] (reproduced in Fig. 3 below). In this section we explain how to perform the calculations for both cases (anticrossing of 2 and of 3 states).

2. *Correction of the effect of the anticrossing of 3 states*

As we can see, the pattern displayed in Fig. 2-right corresponds to the pattern of anticrossing of three states (Fig. 3-right). Hence we will use the program called `FPTripleAnticrossing.f90` (if the anticrossing involves 2 states, we must use `FPDoubleAnticrossing.f90`). This program needs the following input files: i) A file with the parameters for the calculation, called `FPanticrossing.in`; ii) Three files called `eigenvalue1.dat`, `eigenvalue2.dat` and `eigenvalue3.dat`, which contain the information of the eigenvalues vs h as plotted in Fig. 2-right (smooth curves). Note that for that example `eigenvalue2.dat` contains the eigenvalue whose correction is sought (HOMO in this example, gray line in Fig. 2-right), while `eigenvalue1.dat` and `eigenvalue3.dat` contain the eigenvalues which anticross with it (blue and orange lines in Fig. 2-right).

The file with input parameters, `FPanticrossing.in`, can be:

```
  # Input file for the program for the calculation of the dressed
eigenvalues without effects due to anticrossing

  quantity_to_correct = HOMO
  Nh = 33
  PolynomialDegreeg = 6
  Nintervals = 4097
```

where the first variable (`quantity_to_correct`) indicates for which state we calculate the correction (by removal of dressing due to antricrossing effects) and must be either `HOMO` or `LUMO`. The second variable (`Nh`) corresponds to the number of considered values of the displacement parameter (33 in this case, which corresponds to $h = -16, -15, ..., 0, 1, 2, 3..., 16$). The next variable (`PolynomialDegreeg`) corresponds to the degree of the polynomials used in the fitting of the eigenvalue-vs-h curves; such curves are used to find the couplings between the state to analyse (HOMO or LUMO) and the other two states (this coupling is called g in eq. (9) of [1]). The last variable (`Nintervals`) corresponds to the 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`).

The code for the calculation of the corrected eigenvalue can be executed though:

```
$ compile_and_run_find_dressed_wo_Triple_anticrossing.sh
```

or by doing (in terminal):

```
$ rm FPTripleAnticrossing.x
$ gfortran FPTripleAnticrossing.f90 -llapack -o FPTripleAnticrossing.x
$ ./FPTripleAnticrossing.x
```

(note the flag to call `Lapack` [6]). This program consists of several stages:

I) Fitting of the three eigenvalue-vs-h curves stored in `eigenvalue1.dat`, `eigenvalue2.dat`, `eigenvalue3.dat` to continuous, polynomial functions. The shortest distances (d_1, d_2) between the curves of *eigenvalue2* (e.g. HOMO) and the other two eigenvalues (e.g. HOMO-1 and HOMO-2) give the value of the coupling g. Note that `FPTripleAnticrossing.f90` assummes symmetry, hence g is calculated as the average: $g = (g_1 + g_2)/2$, being $g_1 = d_1/2$, $g_2 = d_2/2$.

II) Calculation of the coupling between the two eigenvalues whose correction is not calculated (e.g. HOMO-1 and HOMO-2). This coupling is called g_3 in eq. (9) of [1]. g_3 determines the gap between both eigenvalues at $h = 0$. In our case this gap is zero, as it can be viewed in Fig. 2. We thus calculate the solution to the eigenvalue equation ($\hat{H}v = \lambda v$ with $\hat{H}$ given by eq. (9) of [1], imposing $h = 0$ in it) for different values of g_3 until we find one which makes $v_1 - v_3 \simeq 0$. The chosen eigenvalue solver is `dsbevd`, from `Lapack`.

III) After finding g and g_3, we need to find the parameters Δ and a from eq. (9) of [1]. a is the slope of the oblique asymptotes displayed in Fig. 1-e of [1]; we find it as $a = (E_+(h_{max}) - E_-(h_{min}))/(h_{max} - h_{min})$. Δ is the gap between asymptotes at $h = 0$ (see also that figure); we find it as the slope a times the minimal distance between E_+ (HOMO, in this example) and E_- (or E_0) [7]. It can also be found as: $\Delta = (E_0(h_{min}) + a \cdot |h_{min}|) - E_0(h_{max})$ though we prefer the former method in order to avoid the distorting effect due to the fact that the extreme known values of E_0 do not exactly correspond to the asymptotes. Make sure that Δ is negative if the pattern of the anticrossing is like the one displayed in Fig. 1-e of [1] (if it is flipped, as e.g. Fig. 1 of [7], then Δ must be positive).

IV) At this point, all the parameters of eq. (9) of [1] are known. Hence we solve this equation (again using `Lapack`'s `dsbevd`). This provides the value of the analysed eigenvalue with dressing exclusively due to the coupling between electronic states (and not due to electron-phonon coupling), which we call E_+. This quantity is subtracted to E_+^{DFT}, which is the eigenvalue given by DFT calculations (from `Quantum Espresso`), and contains dressing due to both electron-vibrational and electron-electron couplings. The final output of the program FPTripleAnticrossing.f90 is the difference $E_{+}^{DFT} - E_{+}$ (see Fig. 1-f of [1]). For the example analysed in this section (mode 56 of diamantane with LDA-PZ), this curve is presented in Fig. 4. Note that we took a range of values of h which is low enough to keep the parabolic character of the curve. In Fig. 3 we notice that the fitting of the corrected eigenvalue ($E_{+}^{DFT} - E_{+}$) to a parabola is very good.

V) We make the final correction of the electron-vibrational eigenvalue renormalization. The coefficient of h^2 in the polynomial fitting of the $E_{+}^{DFT} - E_{+}$ function is the concavity of the curve (it is ζ=0.0000425 eV/(a.u.)^2 in Fig. 3). This is what we have to use as concavity in eq. (5) of [1]. Equations (3), (4) and (5) of [1] indicate that the correct renormalization is simply the concavity of the corrected HOMO curve divided by twice the phonon frequency (ω_{ν}, which in this example is 1162 cm^{-1}). In this case, we also need to apply a multiplicative term for conversion between units (1000·219474.6305). This gives a final result for the correct renormalization of the HOMO of diamantane due to the 56th vibrational mode equal to (1000·219474.6305·0.0000425)/(2·1162) = 4.0 meV. This quantity is much lower than the 20.4 meV provided by the original DFT calculation (without correction of anticrossing effects). Hence, we have to correct the result as given by the calculations of Sec. 5 by subtracting 20.4 meV and adding 4.0 meV. This way to proceed must be repeated with all the problematic modes (in this case modes 53, 54, 55, 56, 61 and 62).

Below we display an example of how the output of an appropriate execution of the program looks like.

```
  **************************************************************************
   Now calculating the eigenvalues without the distorting effects of
anticrossings
   ************************************************************************
   The number of h values in the input files is 33
   The degree of the interpolating polynomials to get g is 6
```

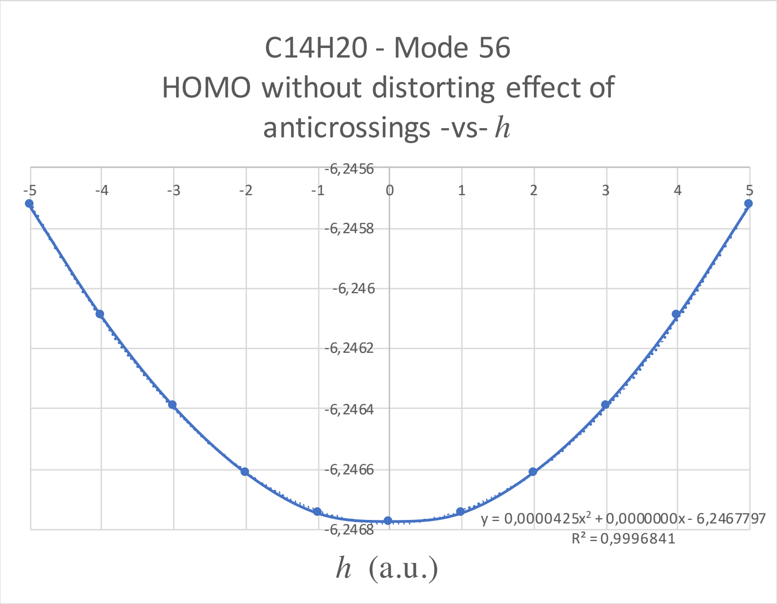

FIG. 4: HOMO without the effect of anticrossing (eV) as a function of the displacement parameter. This is the quantity given as output by `FPTripleAnticrossing.f90`.

```
   ** Now finding the value of g, coupling between the eigenvalue to analyse and the two other eigenvalues which participate in an anticrossing with it **
   Fitting of a polynomial to eigenvalue_1:
   R-squared = 0.99999959679972894
   Fitting of a polynomial to eigenvalue_2:
   R-squared = 0.99999290735785851
   Fitting of a polynomial to eigenvalue_3:
   R-squared = 0.99999947554260271
   The values of the coupling of the three states are:
   g_21:
   The minimum distance is 7.7653751631493684E-002 at x1= 8.0859375000000000 , x2= 8.0859375000000000
   Hence g = 3.8826875815746842E-002
   g_23:
   The minimum distance is 7.7649375503400009E-002 at x1= -8.0937500000000000 , x2= -8.0937500000000000
```

Hence g = 3.8824687751700004E-002

The slope ("a" in eq. (9) of Risueno et al. PRB) of the oblique asymptotes is 8.6432993412017822E-003 eV/(a.u.)

The gap between asymptotes ("Delta" in eq. (9) of Risueno et al. PRB) of the oblique asymptotes is -6.9922941154800355E-002 eV

g_13:

g = 1.7285238050113668E-002

This value of g_13 makes the gap between the two eigenavalues which couple to the eigenv. to analyse (at h=0) be 4.4660607562141408E-007 eV.

--> The HOMO as a function of the displacement parameter (h) WITHOUT the effect of anticrossings is what follows:

```
h, HOMO without distorting effect of anticrossings
-16.0, -6.24330037
-15.0, -6.24313026
-14.0, -6.24304074
-13.0, -6.24304059
-12.0, -6.24313670
-11.0, -6.24333132
-10.0, -6.24361812
-9.0, -6.24398387
-8.0, -6.24440558
-7.0, -6.24485568
-6.0, -6.24530435
-5.0, -6.24572403
-4.0, -6.24609197
-3.0, -6.24639091
-2.0, -6.24661143
-1.0, -6.24674598
0.0, -6.24677360
1.0, -6.24674598
2.0, -6.24661143
3.0, -6.24639091
```

```
   4.0, -6.24609197
   5.0, -6.24572403
   6.0, -6.24530436
   7.0, -6.24485568
   8.0, -6.24440558
   9.0, -6.24398387
   10.0, -6.24361812
   11.0, -6.24333132
   12.0, -6.24313670
   13.0, -6.24304059
   14.0, -6.24304074
   15.0, -6.24313026
   16.0, -6.24330037

   Fit its central region to a parabola using e.g.  Excel or OpenOffice.
To this, take a few values of h; make sure that you take a number of h's
so that it looks still like a parabola.  The concavity of this parabola
(this is its coefficient of h-squared) is what you have to use in eq.
(5) of Risueno et al 2021 to calculate the contribution of the analysed
vibrational mode to the electron-phonon renormalization (discard the
contribution of that mode which appears at the output of frozen-phonon.x,
and use the new one instead).

   ** The calculations finished satisfactorily.  **
```

In the output of the program above, please pay attention to:

- The R-squared of the fittings of the curves (they shoud be above 0.99).
- The values of `g_21` and `g_23`: They should be similar. If that is not the case, perhaps there is an error in your data, or there is not an actual triple anticrossing, or the eigenvalues are not symmetric (with respect to h). In the latter case, this code is no longer appropriate (but a few slight modifications of it would make it usable).

- The "minimum distances" of g_21 and g_23 must be similar (in module). If that is not the case, the system may not be symmetric.

- The final result (i.e. the corrected eigenvalue vs h) should look like a parabola for values of h low enough (like in Fig. 4).

3. Correction of the effect of the anticrossing of 2 states

The calculation of the correction of the effects of an avoided crossing of two states is similar to the case of three states. In our calculations of diamondoids we have noticed that the majority of the anticrossings indeed involve just 2 states. We explain how to perform the calculations through an example, which corresponds to the mode 55 of diamantane (also with frequency 1162 cm^{-1}), analysed with an LDA-PZ functional and a plane wave cutoff of 30 Ry. The way to proceed is very similar to the one presented in the previous section. First of all we fetch the output of the program `GenerateDataToPlotCrossovers.f90` (as presented above). This program works in a very similar manner to its counterparty for 3 states: first fitting the curves of eigenvalue-vs-h to a polynomial, then finding the coupling g as a function of their distance, and finally solving the anticrossing equation (in this case the equation is eq. (7) –(8) indeed– of [1]). If we plot the states 36, 37 and 38 (HOMO-2, HOMO-1 and HOMO) of diamantane as a function of the displacement parameter we obtain the curves displayed in Fig. 5-left. We have to manually change the files so that the curves are smooth (Fig. 5-center). In this case, we notice that one of them (HOMO-2, blue line) is nearly straight and does not present the characteristic shape of anticrossings; hence we can omit it from our calculations (Fig. 5-right).

We include the numerical values of this example in Tab. II, in order to make it possible to check the appropriate working of our programs in all platforms.

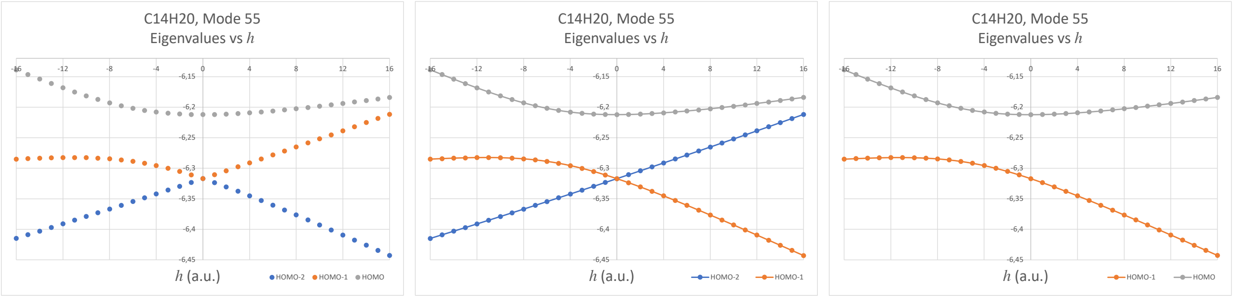

FIG. 5: Eigenvalues of diamantane as a function of the displacement parameter (h) with displacements through the 55th vibrational mode (with frequency 1162 cm^{-1}). The curves on the right side display the typical shape of an anticrossing of two states.

We write the HOMO-1 and HOMO as a function of h in two columns of two files called `eigenvalue1.dat` and `eigenvalue2.dat` (the latter being the HOMO); these files have no header, and the columns are separated by blank spaces, as in the previous section. Before running the calculation script, we need to define a file with its input parameters, `FPanticrossing.in`, which can be as simple as:

```
Nh = 33
PolynomialDegreeg = 6
Nintervals = 4097
```

where the three parameters have the same meaning as their counterparts explained in the previous section (number of h's considered, degree of the polynomial to fit the eigenvalue-vs-h curves to find the minimal distance to calculate the coupling g, and number of values of the grid for such minimal distance calculation).

We then run the script `compile_and_run_find_dressed_wo_Double_anticrossing.sh` or by doing:

```
$ rm FPDoubleAnticrossing.x
$ gfortran FPDoubleAnticrossing.f90 -llapack -o FPTripleAnticrossing.x
$ ./FPDoubleAnticrossing.x
```

If the calculation ran properly, you should see an ouptut like:

h	Eigv.37 (HOMO-1)	Eigv.38 (HOMO)
-16	-6.285160728	-6.138874472
-15	-6.284278366	-6.146532511
-14	-6.283535305	-6.154027879
-13	-6.282967616	-6.161323257
-12	-6.282619119	-6.168373178
-11	-6.28254606	-6.175123842
-10	-6.282812618	-6.18150789
-9	-6.2834929	-6.187452027
-8	-6.284667678	-6.192875573
-7	-6.286417326	-6.197697912
-6	-6.288810831	-6.20184914
-5	-6.291895549	-6.205282208
-4	-6.295685477	-6.207982193
-3	-6.30016165	-6.209968123
-2	-6.305272905	-6.211288445
-1	-6.31095081	-6.212013727
0	-6.317127656	-6.212202682
1	-6.323695849	-6.211984189
2	-6.330616951	-6.21137506
3	-6.337821505	-6.210451971
4	-6.345258242	-6.209266044
5	-6.352886012	-6.207857558
6	-6.36066992	-6.206260383
7	-6.368583013	-6.204501905
8	-6.376602797	-6.202602596
9	-6.384710981	-6.200582184
10	-6.392892602	-6.198454358
11	-6.401135068	-6.196231985
12	-6.409427699	-6.193924033
13	-6.417763687	-6.191540495
14	-6.426132216	-6.189085786
15	-6.434529737	-6.186567413
16	-6.442950727	-6.183990537

TABLE II: Example of eigenvalues vs displacement parameter (diamantane, LDA-PZ, displacement through normal mode 55, whose frequency is 1162 cm^{-1}).

****

Now calculating the concavity of the bare eivenvalue in a 2-state anticrossing

****

```
    The number of h values in the input files is 33
    The number of h values used to solve the threefold anticrossing equation
is 33
    The degree of the interpolating polynomials to get g is 6
    The degree of the interpolating polynomial for the final calculation of
the concavity is 4
    Now calculating the coupling coefficients as a function of the distances
between eigenvalue curves.
    R-squared = 0.99999058920214645
    R-squared = 0.99993193900188015
    The minimum distance is 8.6950942597948000E-002 at x1=
-5.3828125000000000 , x2= -5.3828125000000000
    Hence g = 4.3475471298974000E-002
    Now solving the double anticrossing equation ( between h = -16.00 and h
= 16.00 )
    The bare eigenvalues were calculated.  Find the files to extract the
concavity at eigenvalue1-bare.dat and eigenvalue2-bare.dat.
    ** The calculations finished satisfactorily.  **
```

Make sure that the R^2 of the fitting curves is high (>0.99). If we now plot the bare and dressed eigenvalues we obtain a curve like the one displayed in Fig. 6. In that figure we can notice that the bare eigenvalues could not be calculated at $h = -5$, and that their curves are not-so-smooth near that region. This is because the assumption of double anticrossing that underlies eq. (7) of [1] does not perfectly hold: the results from DFT (from `Quantum Espresso`) do not only have coupling due to electron-electron interaction with one state (which is captured by eq. (7)), but also electron-phonon interaction, potential electron-electron interaction with other states, inaccuracies due to the xc functional, etc. As a consequence, in a few cases (like the one presented here) the value of g calculated is slightly too high, which makes the calculations distort where the coupling effect is maximal. In any case, we are interested in the concavity of the HOMO removing as much as possible

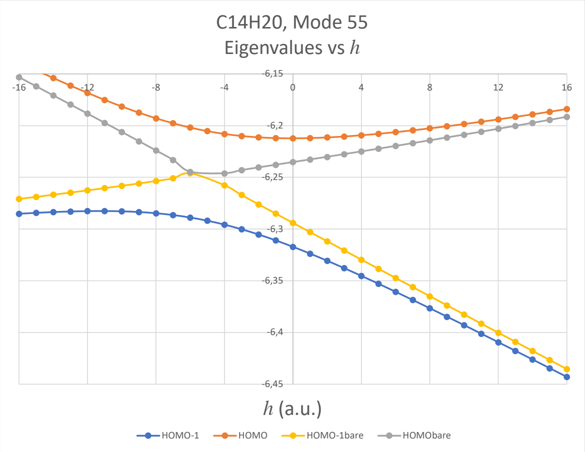

FIG. 6: Bare (corrected, without anticrossing effect) and dressed (from DFT calculations) eigenvalues of diamantane as a function of the displacement parameter (h) with displacements through the 55th vibrational mode (with frequency 1162 cm^{-1}).

from the electron-electron anticrossing (in $h = 0$). The calculated bare HOMO (gray line in Fig. 6) is smooth in the nearby of $h = 0$. We thus select some values of that region (as we did in the case of anticrossing of 3 states) and calculate the concavity. This is summarized in Fig. 7. As we can see in this figure, the fitting of the curve to a parabola has a low concavity ($\zeta = 7 \cdot 10^{-7}$ eV/(a.u.)^2). Using the same formula that we used in the previous section, this leads to a fully negligible value for the frozen-phonon renormalization of the HOMO due to the 55th vibrational mode. This value is 0.066 meV, which contrasts with the value of 20.6 meV obtained from the DFT calculation without correcting the effect of anticrossings. Note that different choices of the range of h used for doing the fitting which provides the concavity may lead to slightly different results; however, all these results are much lower than the uncorrected renormalization value. Hence we should correct the frozen-phonon renormalization of the HOMO of diamantane by adding 0.066 meV and subtracting 20.6 meV to it.

Here ends the explanation on how to perform calculations to correct electron-electron an-

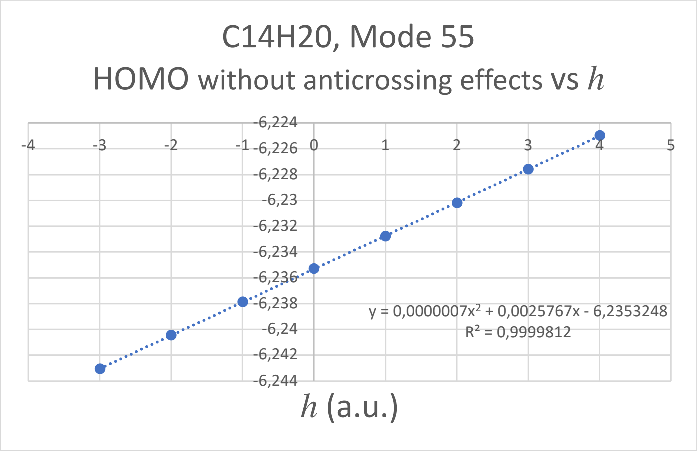

FIG. 7: Bare HOMO of diamantane (i.e. without distorting effects of anticrossing) as a function of the displacement parameter (h) with displacements through the 55th vibrational mode (with frequency 1162 cm^{-1}).

ticrossing effects in the context of calculation of renormalizations due to electron-vibrational interaction. The author of this document is at the disposal of the community for inquiries and discussion. The software explained in this document is freely available for distribution and edition, whenever its authorship (PGR) is acknowledged and the article [1] is cited.

Appendix A: Meaning of the variables of the input file of the program to calculate frozen-phonon renormalizations (`frozenphonon.in`)

In the text below we present an example of an input file (`frozenphonon.in`) for our program to calculate frozen-phonon renormalizations (frozen-phonon.x). An explanation of the meaning of each variable appears above it or on its right.

Temperature = 0.0000001 K # This is the temperature considered in the calculation of the frozen-phonon renormalization; set (nearly) 0 for zero-point renormalization. Erase this field if you perform a calculation with several temperatures (see variables for it below).

Number_of_atoms = 26 # Total number of atoms in the system (26 in $C_{10}H_{16}$)

Number_of_species = 2 # Total number of different atomic species (different chemical elements); for example, 1 in C_{60}, 2 in $C_{10}H_{16}$ 3 in $C_{12}N_2H_{14}$, etc.

# The variable below is the number multiplied by U (the eigenvector of the dynamicalmatrix) to perform the displacements used to do the finite-difference. Make sure that its value is the same which you used in `vib_potPGR.in` to generate the input files with displaced nuclear positions.

Finite_Difference_Parameter = 4.d0 # This is h, the size of the parameter for the calculation of finite-difference displacements.

# The variable below is the name of the output file of a ground state calculation of `Quantum Espresso` (pw.x) storing the electronic eigenvalues of the undisplaced position.

elec_eigval_file_name = `out_scf0.out` # The <Check_crossover> variable below specifies whether or not the possible crossovers in eigenvalues due to the finite-difference displacements are checked and avoided or not. Doing it makes the results much more accurate, but it demands the calculation and printing to file of wavefunctions, which makes calculations slower and demands much disk-storage capability. It is 0 by default, meaning that no overlaps of undislpaced and displaced wavefunctions are evaluated. Set it to 1 to check these overlaps to check the reliability of your calculations. Check_crossover = 0

# The <Crossovers_from_file> variable to 1 means that the program reads the information on crossovers from files (generated previously by making this variable equal to 0). This is useful to avoid the repetition of the slow crossover calculation when evaluating other temperatures. Note that this variable is only useful in the uncommon case that high Finite_Difference_Parameter are used, and that one wants to calculate the renormalizations for several temperatures. Crossovers_from_file = 0

# The <InitialStateFPh> and <FinalStateFPh> variables below make it possible to specify the range of wavefunctions to look for the crossovers (these two variables are optional, 5 states above the LUMO and below the HOMO is taken as default).

# If you want to calculate the HOMO and the LUMO in paralel, submit 2 runs, the first one with the value of FinalStateFPh equal to the highest (maybe degenerate) HOMO index and the second one with the value of InitialStateFPh equal to the lowest (maybe degenerate) LUMO index.

InitialStateFPh = 24

FinalStateFPh = 30

# The two variables below are optional. They specify which phonon modes are considered in the calculations (which can be useful to parallelize the calculations, or to neglect some modes considered not interesting).

InitialMode = 7 # This must be at least 7, for we analyse molecular systems and hence the first 6 frequencies correspond to translations and rotations of the system and they are unphysical

FinalMode = 78 # This is the last index of phononic mode used in the calculations; it must not be higher than 3 times the number of atoms of the system (78 in $C_{10}H_{16}$)

# The variable below is the name of the output file of our program `vib_potPGR.x`. It stores phonon frequencies and the U eigenvectors, which are normalized following the eq. (38) of [5]. You can see the format of `Uvec.dat` at outputfiles-reference

phonon_frequency_file_name = `Uvec.dat`

phonon_frequency_units=cm^{-1} # This is the units used in `Uvec.dat`

# The variables below are the names of the folders storing the outputs of `Quantum Espresso (pw.x)` for ground state calculations with nuclear positions displaced with respect to the relaxed positions (displaced + and - U eigenvectors times <<Finite_Difference_Parameter>>). Note that the names of the files in these folders MUST be "out_scf-modeXXX.out", "eigenval-modeXXX.xml".

folder_output_files_displaced+ = outputsVeffcalc/displaced+

folder_output_files_displaced- = outputsVeffcalc/displaced-

# If the variable below is set to 1, the `.xml` output files of `Quantum Espresso` for the eigenvalues are used for the calculation. These files have many digits and they provide accurate results. If you set this variable to 0 the results will be rather inaccurate.
Eigenvalues_from_xml_files = 1

# The variables below make it possible to impose degeneracy to calculate the renormalizations (see a further explanation below); this can be also be useful for nearly degenerate states.

HOMO_imposed_degeneracy = 2

LUMO_imposed_degeneracy = 2

Band_occupation = 2 # Number of electrons per orbital; by the moment this must be 2, spin is not considered.

To make a sweep of different temperatures, add lines like the following ones to the `frozenphonon.in` file (and remove the <Temperature> variable):

Initial_Temperature = 0.0001d0 K

Final_Temperature = 350.0 K

Number_of_Temperatures = 4

==

The variables of the example above are in general rather intuitive. We shall just explain a bit more on HOMO_imposed_degeneracy and LUMO_imposed_degeneracy. There exist systems whose HOMO and-or LUMO are degenerate. An example of this is C60, where the HOMO has 5-fold degeneracy and the LUMO has 3-fold degeneracy. Ideally, `Quantum Espresso` should provide eigenvalues for HOMO-4, HOMO-3, HOMO-2 and HOMO-1 identical to the HOMO (and LUMO+1, LUMO+2 identical to the LUMO). But in practice this does not always hold (because QE may not impose all the symmetries of the system), and QE may give values for HOMO-4,HOMO-3,HOMO-2,HOMO-1,HOMO differing in a small amount, for example -5.5285, -5.5281, -5.5270, -5.5259, -5.5243 eV, respectively (these quantities differ in less than 4.3 meV). If we perform a frozen-phonon calculation without informing our frozen-phonon.x program that there is indeed a degeneracy, we will obtain the unphysically huge result of a HOMO renormalization of 2817 meV. However, we can avoid this problem by informing the program that there is a 5-fold degeneracy in the HOMO (and a triple degeneracy in the LUMO). To this, we write HOMO_imposed_degeneracy = 5 and LUMO_imposed_degeneracy = 3 in `frozenphonon.in`. Then we run the program and we obtain renormalizations of the HOMO-4,HOMO-3,HOMO-2,HOMO-1,HOMO which cancel each other, and we obtain a final renormalization of the order of meV, as it must be.

[1] P. García-Risueño, P. Han, S. Kumar, and G. Bester, Physical Review B (2021).

[2] R. B. Capaz, C. D. Spataru, P. Tangney, M. L. Cohen, and S. G. Louie, Phys. Rev. Lett. **94**, 036801 (2005).

[3] 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.: Cond. Matter **21**, 395502 (2009).

[4] `https://www.chemie.uni-hamburg.de/institute/pc/arbeitsgruppen/bester/`

`downloads.html`.

[5] S. Poncé, G. Antonius, Y. Gillet, P. Boulanger, J. Laflamme-Janssen, A. Marini, M. Côté, and X. Gonze, Phys. Rev. B **90**, 214304 (2014).

[6] E. Anderson, Z. Bai, and C. e. a. Bischof, LAPACK User's Guide (SIAM, Philadelphia, (1999)), release 3.0 ed.

[7] N. F. Bell, R. F. Sawyer, R. R. Volkas, and Y. Y. Y. Wong, Phys. Rev. A **65**, 042328 (2002).