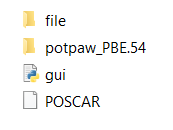
Documentation of the T-dependence spin-resonance codes

1. Framework

The program folder includes two folders and a python program named 'gui.py'. One folder, "file", includes some basic setting files, and the other folder, "potpaw\_PBE.54", includes the pseudopotential files of VASP.



When using the program, one constructs a configuration file "POSCAR" that specifies the atomic configuration of the color center. Then, run the program gui.py, and fill the form in the graphic user interface according to the color center you want to study. The program will generate a folder with a name specified by the user, and the folder includes the necessary files to perform the calculation. Two python programs are generated: 'run.py' and 'analysis.py' to run the VASP jobs and calculate the T-dependence.

The user then copies the folder to a cluster or supercomputing platform with a properly set environment and runs the program' run.py'. This will automatically generate the necessary VASP tasks and collects the output data. After the calculation is done, the program will output a data file named 'out.json'.

Then, the user can run the program 'analysis.py' to derive the temperature dependence of the 2nd-order phonon effects on the spin-resonance shift. The same 'out.json' file can be used for different properties calculated and temperature ranges and can also compute other quantities.

Currently, the program only computes the second-order phonon effect, and the thermal expansion effect needs to be calculated elsewhere and added to the output results. We will update the algorithm later to include the thermal expansion effect and enable the strain dependence calculations.

1. Construct the atomic configuration

The configuration file should be written in a VASP POSCAR format. The atomic configuration includes a supercell of the host material (diamond, for example), including a point defect inside. The lattice constant of the host material will not be changed during the entire calculation, so the user needs to set an accurate value. The atomic configuration of the charged defect will be relaxed, so one does not need to relax the structure in priority. For example, if one uses vesta to construct an NV center structure:

1. get a primitive unit cell of diamond with lattice constant from literature/database

2. Generate a 4\*4\*4 supercell with 128 carbon atoms with vesta or any other atomistic modeling software

3. Replace one C atom with N.

4. Output the file in a VASP POSCAR format.

5. Put the POSCAR file in the folder of the T-dependence program (the same directory with 'gui.py')

Here I emphasize that the computation time increases rapidly with the supercell size. Using a supercell with more than 300 atoms is strongly discouraged. We are updating the feature of using symmetry to reduce the computational difficulty, and this will be realized soon.

1. Generate computational task files

The program 'gui.py' generates the computational tasks. To run this program, one needs a python environment with pymatgen installed. Pymatgen can be easily installed with Anaconda or miniconda. First, create an environment:

conda create --name my\_pymatgen python

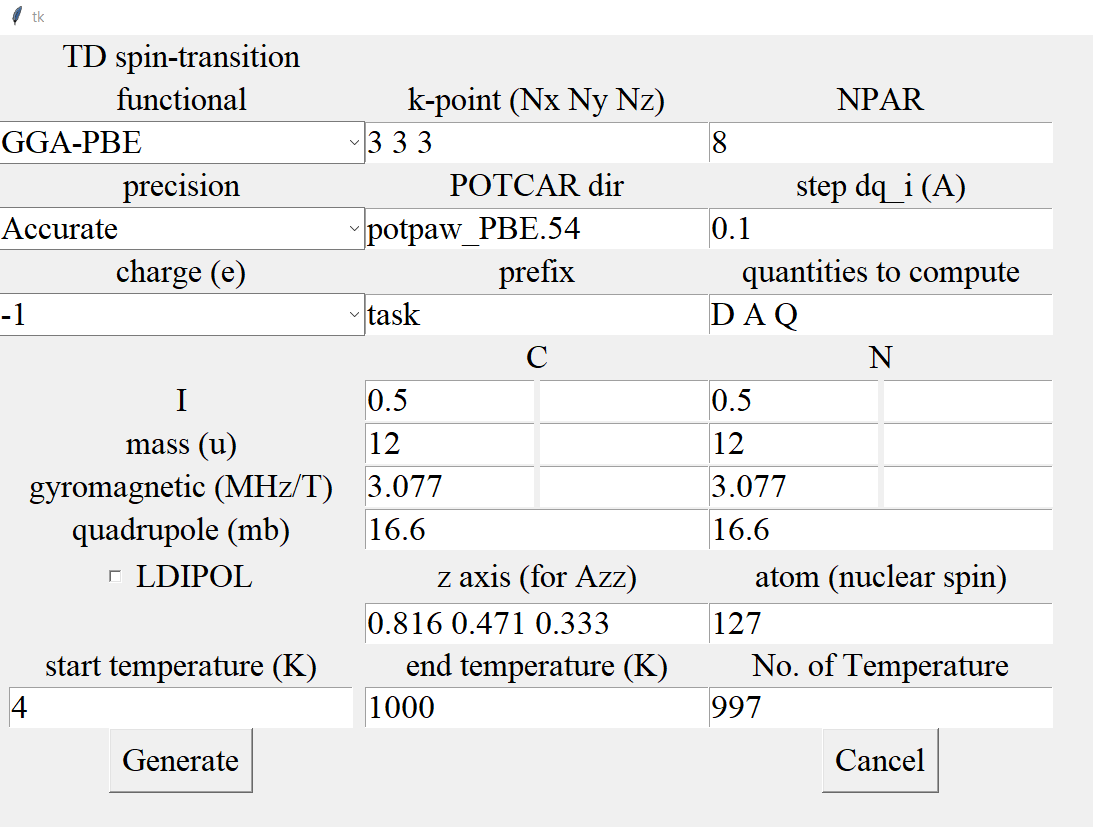
source activate my\_pymatgen *# OSX or Linux*

activate my\_pymatgen *# Windows*

And then run the following command to install pymatgen:

conda install --channel conda-forge pymatgen

Please change the “username” tag in line 16 of the TD\_transition.py script in the file folder to your cluster username (replacing ‘th1543’ by your username on your computational resources). Run program 'gui.py' after putting the POSCAR file in the folder mentioned in the last section. Then you will see a graphic panel as follow:



Then fill out the form according to your specific task. The above is an example for the NV center, and the quantities are explained as follows:

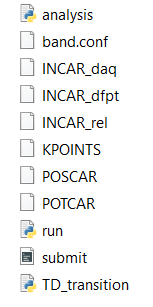
1. Functional: functional used in the VASP calculations when evaluating D, A, and Q. available choice includes GGA-PBE functional, meta-GGA SCAN functional, and HSE06 functional. Atomic relaxation and DFPT calculations will use GGA-PBE functional as default. If you want to change this setting, please manually change the generated INCAR\_rel and INCAR\_dfpt files.
2. K-point setting in all the DFT calculations. The three numbers represent the sampled k points in the a, b, and c directions.
3. NPAR: the parallel processing parameter in VASP in relaxation and D, A, Q calculations. It is recommended to be set as (square root of the number of cores).
4. Precision: the precision setting 'PREC' in VASP. 'Accurate' and 'Normal' are commonly used choices.
5. POTCAR\_dir: The route of the VASP pseudopotential files folder. If you do not need to specify pseudopotential other than the VASP standard PAW potential, leave this as default.
6. Step: the step during the finite differential calculation of the second-order derivatives (Å). The default value is usually acceptable.
7. Charge: the charge state of the color center. For example, -1 means 1 unit of negative charge.
8. Prefix: the name of the folder where we generate the task setting file
9. Quantities to compute: can include D (zero-field splitting), A (hyperfine interaction), and Q (quadrupole interaction)
10. LDIPOL: the dipole correction term in VASP. This corrects the fictitious interaction between an adjacent supercell's charge and dipole moment. However, this sometimes leads to convergence problems for non-orthogonal supercells.

The following tags are only for analysis purposes. After the DFT calculations, one can reset these tags and use the newly generated 'analysis.py' to process the previously output 'out.json' data file. This means that there is no need to redo the DFT calculations to calculate, for example, the nuclear interactions for different atoms or different isotopes:

1. I: the nuclear spin of each element (put in decimal number)
2. Mass: atomic mass of each element (atomic mass unit)
3. Gyromagnetic: gyromagnetic ratio of each element (MHz/T)
4. Quadrupole: quadrupole moment of each element (unit: mb, )
5. z-axis: the z-axis defined for calculating , the hyperfine tensor component in the specified direction.
6. Atom: the serial number of the atom in POSCAR to do nuclear interaction calculations
7. Temperature settings: the temperature points in the output file is a uniform sampling from Tstart to Tend with n points in all. The three quantities are specified by start temperature, end temperature, and No. of temperature, respectively.

The program requires an environment with VASP, python compiler, phonopy, and pymatgen. The VASP tasks are submitted through a user-specified submission file.

After getting everything set, click the button "Generate". Then a folder named as the prefix you specified will be created. The folder includes the following files:



'submit.sh' is an example file to submit the VASP task. Please reset this file according to the requirement of the computation platform you use. Also, copy your specialized submitting file to the 'file' folder, then the program will use this file for every task you set.

If you want some unconventional settings in the VASP calculations, directly modify INCAR\_rel, INCAR\_dfpt, and INCAR\_daq, which specify the settings for the relaxation, DFPT, and D, A, Q calculations, respectively.

1. Run the calculations

Then copy the generated folder to your computation platform. This job is computationally intensive, so please use high-performance computation systems. A typical cost of the calculation is 30,000 core\*hours.

Install the necessary environment in the cluster. Besides VASP, one needs a python compiler with package pymatgen and phonopy. The easiest way is to install via conda. Using the following commands:

conda create --name td\_transition python

source activate td\_transition

*conda install --channel conda-forge pymatgen*

Then follow the instructions to install. After that, use the command.

conda install -c conda-forge phonopy

and follow the instructions to install phonopy. Then, run the program run.py:

nohup python run.py >nohup.out 2>&1 &

The program does a relaxation followed by a DFPT calculation to derive phonon modes. Then, the program begins to sample displacements for the second-order derivatives. It submits 100 jobs into the queue lists, and whenever one job is completed, the program submits another job. After the calculations are completed, the program will output a file named 'out.json' that includes all parameters needed to calculate the temperature dependence.

1. Deriving the temperature dependence

Finally, run the program' analysis.py'. This program reads the file "out.json" and outputs a file named "td\_daq.txt". This file includes the temperature shift and temperature derivative of the calculated quantities (D, A, Q) in the temperature range specified by the user. The file is self-explanatory and can be easily used in plotting figures through softwares like originlab.