Documentation of JDFTx and DMD

Junqing Xu, Kejun Li

January 5, 2023

Contents

1 Workflow of Spin Relaxation Calculation

1.1 Method 1: Spin Relaxation Calculation by JDFTx (Main)

- optimize structure by QE because QE is fast
- optimize structure again by JDFTx
- run scf.in with electronic-scf to obtain self-consistent wavefunction and totalE.in with electronic-minimize to minimize energy
  - electronic-scf and electronic-minimize do the same thing using different algorithm
  - electronic-minimize will be necessary to do if phonon is calculated
  - do scf.in with electronic-scf before totalE.in with electronic-minimize can help efficiently optimize electronic states
- extract polar-correction (LOTO splitting) parameters from QE DFPT calculations (JDFTx cannot provide those parameters)
  - run scf without smearing (if smearing is on, dielectric constant and effective charges will not be calculated)
  - run phonon at q=0 to get dielectric tensor and effective charge
  - do "./readQE.py prefix filename" (e.g. ./readQE.py pwscf totalE) under temp folder to cnovert dielectric constant and effective charge from QE format to JDFTx (total.epsInf totalE.Zeff)
  - **Make sure that the atomic positions in QE are arranged in the same order as the atomic positions in JDFTx**
    - Because the atomic positions are rearranged according to the order of pseudopotentials, specifying pseudopotential in this way "ion-species /PP_path/$ID.upf" can avoid this
- run phonon by part by JDFTx and combine the phonons at all q points
  - if a system is 2D, run the script rotationalCorrection.py to apply rotational correction to interactomic force constant (totalE.phononOmegaSq). File totalE.phononOmegaSqCorr will be generated
  - run PhononDispersion.py to create phonon dispersion data files, phfrq.dat (if no LOTO) and phfrq_loto.dat (if with LOTO)
  - run phonon_bs.plot to plot phonon dispersion and check if there is imaginary frequency
- run wannier until wannier bandstructure compared well with the DFT bandstructure
  - set up the outerWindow the same as eMin and eMax (in Hartree)
  - choose a proper innerWindow to cover both conduction bands and valence bands for test
  - set up wannier centers by either (1) atomic orbitals or (2) random center

  - after wannierization is done, do
    * $ ./WannierBandstruct.py (output wannier.eigenvals)
    * $ ./bandstruct_full.plot (plot overlap of wannier bandstructure and DFT bandstructure)
    * $ ./de.py (for checking bands degeneracy in wannier.eigenvals by taking the energy difference between two wannier bands which are degenerate in DFT. When $\Delta E < 10^{-10}$ eV, the quality of wannier is considered to be good.)
  - to rerun a good wannierization, modify wannier.in by setting up "wannier-minimize nIterations 0 energyDiffThreshold 1e-10" and keep wannier centers the same as they are

- do lindbladInit to get wanted interactions, e.g. e-ph, e-e, e-imp, in a dense **k** grid
- do real-time evolution of spin (density matrix dynamics, DMD) to get the spin dynamics, and spin lifetime can be obtained by fitting the spin dynamics data

1.2 Method 2: Spin Relaxation Calculation by QE+JDFTx

1.2.1 Part 1: prepare wfc and phonon by QE

- optimize structure by QE in a **k** grid ($k_1 \times k_2 \times k_3$)
  - check the line with "FFT dimensions" in QE output
  - run JDFTx nscf which is totalE.in and check the line with "fftbox"
  - make sure that ecutwfc in QE can generate a FFT of size as large or equal as the minimum FFT in JDFTx to avoid size mismatch of totalE.wfns in wavefunction convertion from QE to JDFTx
- separately calculate phonon for each q point in a coarse **q** grid ($q_1 \times q_2 \times q_3$) smaller than or equal to **k** grid, and **k** grid and **q** grid need to be commensurate. In this step, one will get files suffixed with dvscf_qX which contain the lattice-periodic variation of the self-consistent potential at qX. The dvscf file will be used in e-ph calculation
- combine the phonons from all the q points and run q2r to get interatomic force constant in q space (a binary file named after force_constants_q.dat which is generated by using Junqing's modified QE-6.6)
- plot phonon dispersion to check whether or not there is imaginary frequency
- go into folder "epw", run scf, nscf and epw in the **q** grid to get the file suffixed with epbX which contains the Hamiltonian, the dynamical matrices and the electron-phonon matrix elements on the initial (coarse) k and q grids in the full Brillouin zone. The **k** and **q** grids in the inputs are set up as follows
  - scf.in: K_POINTS automatic
  - nscf.in: K_POINTS crystal (this k mesh can be generated by using wannier90 utility kmesh.pl)
  - epw.in: at the end of the input file, specify the **q** grid from file dyn0

1.2.2 Part 2: wfc conversion

- convert QE wfc to JDFTx wfc
  - go into folder "qe", run scf and nscf by QE in the **k** grid with the same nbnd as that in epw
  - go into folder "jdftx", do a dry run for the JDFTx scf in the same **k** grid with the same nbnd and same ecutwfc to get totalE.sym, totalE.kPts and totalE.Gvectors for format conversion by
    $DIRJDFTX/jdftx -ni totalE.in > totalE.out
  - go to the parent directory and use "convert_twoqe_jdftx.py" to convert QE wfc to JDFTx wfc
- do nscf and bandstructure by JDFTx to get totalE.eigenvals, totalE.eigstates, totalE.out, totalE.S, totalE.Vscloc and bandstructure
  - make sure "dump End State" is not included in totalE.in to avoid the output of new totalE.wfns

1.2.3 Part 3: eph matrix conversion

- do phonon dry run by JDFTx to get the necessary files (e.g. phonon.out) for checking qmesh in JDFTx is the same as that in QE
  - this calculation may require large memory
  - reduce the –ntasks-per-node, i.e. reduce the number of cores to allocate more memory per node for this calculation
- convert e-ph matrix from EPW to JDFTx
  - go to the parent directory
  - $ python3 read_epw.py epw/temp/ prefix jdftx/ | tee log (Sec. A)
  - get the following files which are soft linked in folder "jdftx"
    - totalE.phononBasis (atomic species and positions in a unit cell, in folder "phonon")
    - totalE.phononCellMap (phonon cell map for supercell method, in folder "phonon")
    - totalE.phononCellWeights (binary file, in folder "phonon")
    - totalE.phononOmegaS (inter-atomic force matrix, in folder "phonon")
    - totalE.phononHsub (e-ph matrix, binary file, in the parent folder)
    - force_constants_q_cpp.bin
    - ph_freq_of_first_two_q_in_mev.out
  - if 2D system, apply rotation correction to force constant by
    - $ python3 rotationalCorrection.py

    else pass
  - check the the e-ph matrix element in log to make sure no unusual (extremely large) e-ph matrix element

- do polar correction (LOTO splitting) to phonon for the following JDFTx calculations (JDFTx does not have polar correction)
  - run scf without smearing
  - run phonon at q=0 to get dielectric tensor and effective charge
  - do "./readQE.py prefix filename" (e.g. ./readQE.py pwscf totalE) under temp folder to cnovert dielectric constant and effective charge from QE format to JDFTx (total.epsInf totalE.Zeff)

1.2.4 Part 4: wannierization

- run wannier by JDFTx until wannier bandstructure compared well with the DFT bandstructure
  - set up the outerWindow the same as eMin and eMax (in Hartree)
  - choose a proper innerWindow to cover both conduction bands and valence bands for test
  - set up wannier centers by either (1) atomic orbitals or (2) random center
  - after wannierization is done, do
    - $ ./WannierBandstruct.py (output wannier.eigenvals)
    - $ ./bandstruct_full.plot (plot overlap of wannier bandstructure and DFT bandstructure)
    - $ ./de.py (for checking bands degeneracy in wannier.eigenvals by taking the energy difference between two wannier bands which are degenerate in DFT. When $\Delta E < 10^{-10}$ eV, the quality of wannier is considered to be good.)
  - to rerun a good wannierization, modify wannier.in by setting up "wannier-minimize nIterations 0 energyDiffThreshold 1e-10" and keep wannier centers the same as they are

1.2.5 Part 5: real-time spin dynamics

- do lindbladInit to get wanted interactions, e.g. e-ph, e-e, e-imp, in a dense **k** grid
- do real-time evolution of spin (density matrix dynamics, DMD) to get the spin dynamics, and spin lifetime can be obtained by fitting the spin dynamics data

2 Input Parameters

2.1 lindbladInit.in

Table 1: Parameters in lindblad initialization input lindbladInit.in.

parameter	input	meaning
NkMult	an integer, e.g. 16	The number of times that the k-point sampling increases in each cartesian direction. Finally the factor is NkMult^3.
dmuMin	a number, e.g. -0.5	Minimun chemical potential for hole in the unit of eV, the code does dmuMin = min{dmuMin,VBM}. Fermi level is set to 0.
dmuMax	a number, e.g. 0.5	Maximun chemical potential for electron in the unit of eV, the code does dmuMax = max{dmuMax,CBM}. And this is the chemical potential at which spin mixing $\|b\|^2$, spin lifetime by rate equation are evaluated at LindbladInit. Fermi level is set to 0.
Tmax	an integer, e.g. 300	Temperature in the unit of K at which a calculation is performed
nkBT	an integer, e.g. 7 (default)	This parameter controls energy range (Emargin = nkBT * Tmax). k points sampled in the energy range are used.
Bext	three numbers separated by ",", e.g. 0,0,1	External magnetic field (B_x, B_y, B_z) in the unit of Tesla. Include this parameter when calculating spin dephasing.
EzExt	a number, e.g. 5	External electric field along z-axis in the unit of V/nm.

enforceKramerDeg	yes or no	Bands that are degenerate in DFT may be off from degeneracy because wannierization may not 100% reproduces DFT bandstructure. This parameters force the bands to be degenerate by taking the average value.
kparis_eph_eimp	1 or 0	When selecting the k-pairs, if kparis_eph_eimp is 1, both e-ph and e-impi scatterings are considered. Otherwise, only e-ph scattering is connsidered.
read_kpts	1 or 0	Choose to read kpts from previous database
read_kpairs	1 or 0	Choose to read kpairs from previous database
read_gfack	1 or 0	Choose to read g-factor from previous database
gfac_mean	three numbers separated by ",", e.g. 2.00232,2.00232,2.00232	g factor in cartesian coordinates (g_x, g_y, g_z). This is ignored when "orbitalZeeman yes"
writeU	1 or 0	Write the eigenvectors of wannier Hamiltonian for analysis.
band_skipped	an integer	Starting band index of wannier relative to DFT, can be used to determine VBM and CBM correctly in case VBM is not zero (metal, Fermi smearing or finite electric field). This parameter is automatically determined by the code now.
pumpOmegaMax	a number, e.g. 0.0	Maximum pump energy in the unit of eV. Zero means pump is not active.
pumpTau	a number, e.g. 10000	Pump pulse time width in the unit of fs.

probeOmegaMax	a number, e.g. 0.0	Maximum probe energy in the unit of eV. Zero means pump is not active.
ePhOnlyElec	1 or 0	If 1, ePhOnlyHole will be 0, and only e-ph of conduction bands will be written down. If 0, ePhOnlyHole will be 1, and only e-ph of valence bands will be written down.
ePhOnlyHole	1 or 0	If 1, ePhOnlyElec will be 0, and only e-ph of valence bands will be written down. If 0, ePhOnlyElec will be 1, and only e-ph of conduction bands will be written down.
ePhMode	DiagK or Off	If Off, electron-phonon scattering will be turned off. If DiagK, it will be turned on.
modeStart	an integer, e.g. 0	The phonon mode from which start for electron-phonon scattering. If 0, start from the first phonon mode.
modeStop	an integer, e.g. -1	The phonon mode with which end for electron-phonon scattering. If -1, end with the last phonon mode.
detailBalance	1 or 0 (default)	"1" means the detailed balance technique is used to deal with the energy conservation for electron-phonon and electron-impurity scattering. "0" means this technique is not used. Usually this technique gives more physical real-time dynamics results.
ePhDelta	a number, e.g. 0.02	Gaussian smearing in the unit of eV for electron-phonon scattering.

nEphDelta	an integer, e.g. 1000	A parameter for selecting the k-pairs. For scattering between two electronic state kn and k'n', if $\|E_{kn}$ - $E_{k'n'}\|$ <= ePhDelta * nEphDelta, the energy conservation is satisfied and the k-pair k-k' will be considered in later real-time calculations.
keepgm	1/0	This parameter controls whether or not to output electron-phonon matrix element data in the folder /ldbd_data, then DMD code can output electron-phonon matrix as a function of q.

PS:

- The lindbladInit code can roughly output some properties (carrier density, spin mixing etc.) for an estimation of spin lifetime. All of these properties can be found in the real-time dynamics simulation output.
- Properties like spin mixing from lindbladInit is calculated at Fermi level, which is set to be dmuMax. To get all the properties of hole from lindbladInit, set dmuMax equal to dmuMin.

2.2 DMD input param.in

Table 2: Parameters in DMD input param.in for task control and algorithms.

parameter	input	meaning
restart	1 (default) or 0	If 1, start from scratch; else if 0, restart.
alg_scatt	lindblad (default) or conventional	It determines the form of the scattering term of the master equation.

alg_eph_sepr_eh, alg_eph_need_elec, alg_eph_need_hole	1 or 0	If alg_eph_sepr_eh = 0, electron and hole will be calculated together. If alg_eph_sepr_eh = 1, electron and hole will be calculated separately. Then, if "ePhOnlyElec 1" is in lindbladInit.in, set alg_eph_need_elec = 1 and alg_eph_need_hole = 0. If "ePhOnlyHole 1" is in lindbladInit.in, set alg_eph_need_elec = 0 and alg_eph_need_hole = 1.
alg_sparseP	1 or 0 (default)	If 1, the code will convert the generalized scattering-rate matrix P to a sparse matrix. If 0, matrix P is kept dense. Search for "ns_tot" in output to see how many elements of P_i matrices are larger than several thresholds (default 1e-40) generally, larger energy range and smaller smearing make P_i more sparse.
alg_phenom_relax	1 or 0 (default)	If 1, phenomenon relaxation $\dot{\rho} = -(\rho - \rho_{\mathrm{eq}})/\tau_{\mathrm{phenom}}$ will be turned on. If 0, it will be turned off.
Bxpert, Bypert, Bzpert	a number, e.g. 0.0 (default)	Magnetic field perturbation, one of the mothods that perturbs a system to generate spin unbalance, in the unit of Tesla. x, y and z are cartesian coordinates consistent to input cell parameters.

pumpMode	coherent or lindblad or perturb	Perturbation by pumping, the other method that is used to perturb a system to generate spin unbalance. The pump pulse is Gaussian. coherent : real-time pump using coherent formula $-i[\mathrm{P}(t), \rho]$. lindblad : real-time pump using lindblad formula. perturb : generate an initial pump perturbation.
pumpA0	a number, e.g. 0.0	Pump amplitude, pump power = (pumpE * pumpA0)^2 / $(8\pi\alpha)$
pumpPoltype	LC or RC or Ex or Ey	Pump polarization type.
pumpE	a number, e.g. 0.1	Pump energy in eV.
pumpTau	a number, e.g. 50	Pump pulse width in the unit of fs. This introduces a weight function $\exp(-t^2/2\tau^2)/\sqrt{\sqrt{\pi} * \tau}$ into pump amplitude
probePoltype1	LC or RC or Ex or Ey	The first probe polarization type.
probePoltype$i$	LC or RC or Ex or Ey	The i-th probe polarization type.
probeEmin, probeEmax	a number	These parameters specify the probe energy range in the unit of eV.
probeDE	a number, e.g. 0.01	Probe energy step in the unit of eV.
probeTau	a number, e.g. 2000	Probe pulse width in unit the unit of fs.
t0	a number, e.g. 0 (default)	The initial time of spin dynamics in the unit of fs.
tend	a number, e.g. 1e6	The end time of spin dynamics for reporting in the unit of fs. This number needs to be larger at lower temperature because spin dynamics is slower at lower temperature. 1e6 is used for T=4K.

tstep	a number, e.g. 1e3	Time step in the unit of fs.
tstep_pump	a float	Time step for reporting during pump (pump time center ±6*pumpTau), in the unit of fs.
freq_measure_ene	a integer	This parameter specifies how often to print energy-resolved observables.
temperature	a number, e.g. 300	Temperature at which calculation is done. Currently, it is actually directly read from the output of the initialization calculation, so that it is not necessary to set temperature.
mu	a number, e.g. 0.5	Chemical potential in the unit of eV, usually relative to VBM, must be in range [dmuMin, dmuMax] in initialization
carrier_density	a number, e.g. 1e18	Carrier density in the unit of cm^{-d}, where d (3,2,1,0) is the dimension of the material. Positive means electron density and negative means hole density. It can be referred to and smaller than the carrier density in the lindblad initialization output lindbladInit.out. When this parameter is used, mu is ignored.
tau_phenom	float	Phenomenon relaxation time in the unit of ps.
scrMode	medium or none (default)	"none" means no screening will be computed. "medium" means the dielectric function = interband dielectric constant * intraband dielectric function. interband one must be provided by epsilon_background. intraband one will be computed by the program.

epsilon_background	a number, e.g. 30	static dielectric constant ε_0.
eeMode	Pee_update or Pee_fixed_at_eq	The electron-electron scattering matrix P depends on the density matrix ρ. There are options of updating the scattering matrix P^{e-e} (Pee_update) and fixing the matrix (Pee_fixed_at_eq) during spin dynamics.
freq_update_ee_model	integer	This parameter controls how often P^{e-e} is updating duing spin dynamics. If <= 0, P^{e-e} is updated every step.
alg_eph_enable	1 (default) or 0	If 1, enable the electron-phonon scattering. If 0, diable electron-phonon scattering.
alg_only_eimp	1 or 0 (default)	If 1, only consider the electron-impurity scattering. If 0, consider the scattering processes other than electron-impurity.
impMode	ab_neutral or model_ionized	Currently, the electron-impurity scattering by neutral defects is computed by supercell method at DFT, and that by charge defects is computed by a charge defect model. If ab_neutral, electron-impurity scattering considers short-range interaction due to neutral defects. If model_ionized, the scattering considers long-range interaction due to charge defects.
impurity_density	a number, e.g. 1e18	Impurity density in the unit of cm^{-d}, where d (3,2,1,0) is the dimension of the material.

3 Additional Calculation Procedure

3.1 Spin Lifetime under Magnetic Field

- add "saveRP yes" to wannier.in and rerun wannier with "nIterations 0" (R stands for electron position, P stands for momentum)
- add "Bext B_x, B_y, B_z" and "orbitalZeeman yes" to lindbladInit.in
- run initialization and real-time dynamics

3.2 Postprocessing: get e-ph matrix element involved in spin-flip process and spin-conserving process

- Run lindbladInit with "keepgm 1" in lindbladInit.in to save the e-ph matrix into files
- Enter folder ldbd_data, find the line with "modeStart modeStp modeSkipStart modeSkipStop" and change -1 to 0 for modeSkipStop
- Run DMD with the same number of CPUs as lindbladInit. This step will read the e-ph matrix elements and analyse them.
- Enter folder eph_analysis
  - gsf2q_*.out is for spin flip
  - gsc2q_*.out is for spin conserving
- gsf2q and gsc2q are sum of e-ph matrix elements as a function of q

A Appendix: Parameters in read_epw.py

read_epw.py

- reads dynamical matrix (dynq) and eph matrix (eptmp) from files *.epb that are output from the modified elphon_shuffle_wrap.f90 in QE-6.6
- reads force constant in q space (self.ifcq) from file "force_constants_q.dat" that is from modified phonon in QE-6.6, and outputs file "force_constants_q_cpp.bin" and file "totalE.phononOmegaSq"
- diagonalizes self.ifcq to get phonon eigenvalues (w_qe) and eigenvectors (u_qe) from QE
- diagnoliazes totalE.phononOmegaSq to get phonon eigenvalues (w_jd) and eigenvectors (u_jd) from QE-JDFTx interface
- diagonalizes dynq to get phonon eigenvalues (w) and eigenvectors (u) from EPW
- converts eptmp to JDFTx compatible file "totalE.phononHsub"

Table 3: The structure of the epw output file (suffixed with epb) from Junqing's modified code, where the line "WRITE(iuepb) nbnd, nks, nmodes, nqc1, nqc2, nqc3, xqc, dynq, epmatq" is added. Number of q points nq=nqc1*nqc2*nqc3.

parameter	type	type in python	size (bytes)	meaning
head	int	int32	4	the start of file *.epb
nbnd	int	int32	4	number of bands
nks	int	int32	4	number of k points
nmodes	int	int32	4	number of phonon modes
nqc1, nqc2, nqc3	int	int32	4*3	coarse **q** grid
xqc	float	float	8*3*nq	the coordinates of **q** grid
dynq	complex	complex_	16*nmodes *nmodes*nq	dynamical matrix
epmatq	complex	complex_	16*nbnd*nbnd *nks*nmodes*nq	electron-phonon matrix at q
tail	int	int32	4	the end of file *.epb

B Appendix: spin_dft.py

For checking spin mixing at DFT. After running this script, run fitQuality.py in folder Wann to estimate the wannierization quality.