Manual

Introduction

This program collects algorithms for determining lattice oscillations. The frequencies of lattice oscillations are important quantuties in several areas of solid state physics. Different approaches can be used to describe lattice oscillations, as the dynamic structure factor (DSF), velocity autocorrelation function (VACF), and projected velocity autocorrelations in **q**-space. Or the dynamic structure factor can be projected onto a set of eigenvectors. The projected velocity autocorrelation (PVACF) is a projection of the VACF onto a set of eigenvectors. The eigenvectors can for example be taken from harmonic PhonoPy calculations. For details about the implemented methods see [1]

Installation in UNIX

To install the DSLEAP in UNIX unpack the tarball tar -xvzf DSLEAP.1.tar.gz. Then go to the main folder cd DSLEAP. Then export the desired gnu compiler export FC=gfortran. Type cmake -DCMAKE_BUILD_TYPE=Release CMakeLists.txt. Then type make and the executable will be found in ./bin/DSLEAP

Table 1: General MD spec flags

Step	command
(1)	tar -xvzf DSLEAP.tar.gz
(2)	cd DSLEAP
(3)	export FC=gfortran
(4)	cmake -DCMAKE_BUILD_TYPE=Release CMakeLists.txt
(5)	make
(6)	find executable in ./bin folder

General settings and the Phonon.in file

There is a set of general settings for the phonon computation. These flags will specify the conditions of the molecular dynamics (MD) simulation which is analyzed. The flags shown in table are specified in the **Phonon.in** file

[1] Lahnsteiner et.al., Anharmonic phonon properties for finite temperature materials by machine-learning force-fields and the breakdown of the phonon quasiparticle picture for $CsPbBr_3$

Table 2: General MD spec flags

flag	meaning	type	default value
NSTART	structure in file to start sampling	integer	1
NEND	total number of structures in XDATCAR file	integer	-
DSTEP	analyze every $DTSEP^{th}$ structure	integer	1
TSTEP	time step in what unit ever	float	1.0
NX	number of periodic images in x	integer	-
NY	number of periodic images in y	integer	-
NZ	number of periodic images in z	integer	-
DSTC	time window for average correaltion functions	integer	NEND
DYNSTRUC	True/False switch on/off Dynamic structure factor	bool	False
PVACF	on/off projected VACF on eigenvectors and **q**-space	bool	False
PVACK	on/off projected VACF **q**-space only	bool	False
PROJFAC	on/off projected DSF on eigenvectors	bool	False

Specifing one of the methods with DSTC, DYNSTRUC, PVACF, PVACK or PROJFAC requires different input files. The different methods and their usage will be described in the following sections. The code also supports openmp parallelization. If you do not want the program to use all your cores do export OMP_NUM_THREADS=# of cores before execution.

Dynamic structure factor computation

To compute the DSF switch on the flag DYNSTRUC=True. And set the other remaining parameters defined in the Phonon.in file. To compute the DSF you have to specify a **q**-grid. This can be done by supplying a q-points file called "QVectors.in" containing on the first line an integer with the number of **q**-points. Then the **q**-points are listed in a 3 column format. If no "QVectors.in" file is supplied the default **q**-points of the commensurate **q**-grid are used. As output you will obtain a file called DynamicStructureFactor.out containing the DSF. The first column is the frequency axis. The frequency is inverse time units of the time step you supplied. In the following columns the signals for the different **q**-points are listed. The file StructureFactor.out contains the static structure factor. The first column is the Qpath. And last the file StructureFactor_vs_t.out contains the structure factor in the time domain. The first column of the file is the time axis.

The velocity autocorrelation function in q space

To compute the **q** space projection of the velocity autocorrelation function set the flag PVACK=True. This method needs a masses.in file described in section 1.9. The masses are needed for weighing the velocities properly. For this method the atoms have to be assigned to the unitcells in which they are located. This can be done with the BoxList.in file described in section 1.8. As output you are going to obtain numberd PVACF_KO.out{II} files where II is an index from 1 up to the number of atoms per unit cell. The first column of this file contains the frequency grid and the following columns contain the signals for the used **q**-vectors. So a **q**-space projected velocity autocorrelation function is computed for every atom in the unitcell. The frequency grid is given in inverse time units as defined in the Phonon.in file. In the files PVACF_KO_vs_t.out{II} you will find the time signals. These have the time axis on the first column and then the signals for the different **q**-values follow.

Projected velocity autocorrelation PVACF

To compute the projected velocity autocorrelation function set the flag PVACF=True. This computes a PVACF projected onto a set of basis vectors and onto a set of **q**-vectors. The routine needs the input files BoxList.in, masses.in and BasisVector.in files. If no BasisVector.in file is supplied the code will try to find a QVectors.in file. If this file is found the **q**-vectors from this file are used and a diagonal basis. If no QVectors.in file is supplied the default **q**-vectors will be used in combination with the diagonal basis. The masses.in file is needed to weigh the velocities with their atomic masses and the BoxList.in is needed for the spatial fourier transforms. As output you will obtain PVACF.out{II} files. These files will contain the frequency axis on the first column and the signals for every **q**-vector on the following columns. Such a file will be written for every phonon branch denoted by the counter {II}. The files PVACF_vs_t.out{II} will contain the corresponding time signals with the time axis on the first column.

Projected Dynamic Structure factor

To compute the projected dynamic structure factor set the flag PROJFAC=True. This computes a DSF projected onto a set of basis vectors and onto a set of **q**-vectors. The routine needs the input files BoxList.in and BasisVector.in. If no BasisVector.in file is supplied the code will try to find a QVectors.in file. If this file is found the **q**-vectors from this file are used and a diagonal

basis. If no QVectors.in file is supplied the default **q**-vectors will be used in combination with the diagonal basis. As output you will obtain Projected-DSF.out{II} files. These files will contain the frequency axis on the first column and the signals for every **q**-vector on the following columns. Such a file will be written for every phonon branch numbered by {II}. The files StructureFactorProj_vs_t.out{II} will contain the corresponding time signals with the time axis on the first column.

Input file BoxList.in

The BoxList.in file contains a connection table assigning every atom to a unit cell building up the super cell. The file contains a table where the number of lines are the number of unitcells building up the supercell. Every line contains N integer numbers, where N defines the number of atoms per unitcell. The numbers represent the atom number in the XDATCAR file. This file is needed for PVACF and PVACKO.

Input file masses.in

The masses.in contains the number of atoms in the unit cell on the first line. The following lines contain a single number describing the mass of the atoms in the unit cell. The order has to be the same as in the XDATCAR file. This file is needed for PVACF and PVACKO.

The BasisVector.in file

This file is needed when computing PVACF or PROJFAC. The projected velocity autocorrelation function in **q**-space and onto a set of basis vectors. The PROJFAC is a projection of the dynamic structure factor onto the same set of eigenvectors. The file contains 4 integer numbers on the first line. The first number is the number of different **q**-points. The second number is the number of branches per **q**-point. The third number defines the number of atoms in the unit cell. And the last number defines the dimensionality of the underlying space, which is always 3. The file is then structured as follows. The next line contains the first **q**-vector. Then without an empty line in-between the first phonon branch of **q**-vector one follows. Then an empty line and the next phonon branch follows. The number of lines per branch is the number of atoms per unit cell. Like this all branches for the first **q** point are listed. Then 2 empty lines follow and the next **q**-vector is defined. Followed by the first phonon branch. Then followed by the eigenvectors of the other branches separated by single empty line.

Example with input files

In the folder EXAMPLE you can find a XDATCAR file corresponding to a microcanonical ensemble simulation of the $CsPbBr_3$ perovskite. The XDATCAR contains a $10 \times 10 \times 10$ unit cell with 20000 MD steps. The simulation was done at 400K with a time step of 10fs. The folder EXAMPLE also contains a file unitcell.vasp containing a POSCAR file format of a 5 atom cubic unit cell that was replicated for the MD simulation. Then there is a folder ./PC/EXAMPLE/testRun. This folder contains all the relevant data to try the above mentioned implementations of the lattice dynamics methods. You just have to unpack the XDATCAR.tar.gz file with ***tar -xvzf XDATCAR.tar.gz***. Then ***cd testRun***. Then you can manipulate the Phonon.in file. Afterwards execute ***../../bin/DSLEAP*** and check your output files.