LetzElPhC Documentation

Muralidhar Nalabothula

March 14, 2024

Contents

Chapter 1

About the Code

LetzElPhC is a C code, which computes electron-phonon coupling matrix elements from the outputs of standard DFT and DFPT calculations. It currently only supports the Quantum Espresso code with plans to extend it to the Abinit code. The primary objective of this project is to support electron-phonon related calculations in the YAMBO code (>=5.2) and works only with *norm-conserving* pseudo-potentials. It is currently released under MIT license and is hosted on github.

1.1 Main features

- Exploits full crystal symmetries and the output quantities are fully compatible with the YAMBO code without any phase issues.
- Has multiple levels of parallelization: OpenMP, plane-wave, k-point and q-point.
- Fully parallel IO using parallel NetCDF-4/HDF5 libraries.
- Highly portable. The code can be compiled more or less on any CPU architecture and operating system with minimal/no changes.

Chapter 2

Installing the Code

2.1 Mandatory requirements

- GNU Make
- Your favourite C99 compiler with complex number support.
  Ex : GCC, Clang, ICC, AMD C-Compiler, MinGW (for windows), PGI, Arm C compilers etc.
- MPI implementation (must support at least MPI-standard 2.1 standard)
  Ex: Open-MPI, MPICH and its flavours, Intel MPI compiler, Microsoft MPI (for windows) etc.
- FFTW-3 or Intel-MKL
- HDF5 and NETCDF-4 libraries with Parallel IO support
  (must be compiled with MPI)
- Your favourite BLAS library.
  Ex : Openblas, Blis, Intel-MKL, Atlas etc.

2.2 Installing

LetzElPhC employs standard make build system. There are some sample make files in *sample_config* directory. Copy it to the *src* directory and rename it as *make.inc*. Now, go to the *src* folder and edit the *make.inc* file according to your needs and type (in the same *src* directory)

```
$ make
#### You can also compile the code in parallel with -j option
$ make -j n
```

```
#### where n is number of processess.
```

If you successfully compile the code, you should find "***lelphc***" executable in the *src* directory.

If you have hard time finding the libraries, you can go the YAMBO code installation directory, and open the *report* file in the *config* directory (of course after successfully installing YAMBO). This will list all the required libraries and include paths.

Here are the list of variables in the *make.inc* file with explanations.

```
CC                  :=  mpicc
#### MPI C compiler mpicc/mpiicc (for intel),
CFLAGS              := -O3
#### -O3 is to activate compiler optimizations
LD_FLAGS            :=
#### use this to pass any flags to linker

#### **** OPENMP BUILD ***
#### If you wish to build the code with openmp support
#### uncomment the below line
# OPENMP_FLAGS      := -DELPH_OMP_PARALLEL_BUILD
#### Aditionally, you need to add openmp compiler flag to
#### CFLAGS and LD_FLAGS.
#### Just uncomment the below two lines
# CFLAGS            += -fopenmp ## use -qopenmp for intel
# LD_FLAGS          += -fopenmp ## use -qopenmp for intel

#### FFTW3 include and libs (see FFT flag in yambo config/report)
FFTW_INC            :=  -I/opt/homebrew/include
FFTW3_LIB           :=  -L/opt/homebrew/lib -lfftw3_threads -
   lfftw3f -lfftw3f_omp -lfftw3_omp -lfftw3
#### Note if using FFTW
#### Yambo uses double precision FFTW regardless of the precision
   with which Yambo is built. In contrast, you need to link single
   (double) precision FFTW for single (double) precision LetzElPhC.
    please refer to https://www.fftw.org/fftw3_doc/Precision.html .
    Also you refer to https://www.fftw.org/fftw3_doc/
   Multi_002dthreaded-FFTW.html  if compiling with openmp support.



#### Blas and lapack libs (see BLAS and LAPACK flag in yambo config
   /report)
BLAS_LIB            :=  -L/opt/homebrew/opt/openblas/lib -lopenblas
#### you need to add both blas and lapack libs for ex : -lblas -
   llapack

#### netcdf libs and include
#### (see NETCDF flag in yambo config/report)
NETCDF_INC          :=  -I/Users/murali/softwares/core/include
NETCDF_LIB          :=  -L/Users/murali/softwares/core/lib -lnetcdf

#### hdf5 lib (see HDF5 flag in yambo config/report)
```

```
36 HDF5_LIB             :=  -L/opt/homebrew/lib  -lhdf5
37
38 #### incase if you want to add additional include dir and libs
39 INC_DIRS             :=
40 LIBS                 :=
41
42
43 #### Notes Extra CFLAGS
44 ### add -DCOMPILE_ELPH_DOUBLE if you want to compile the code in
       double precession
45 ### if you are using yambo <= 5.1.2, you need to add "-
       DYAMBO_LT_5_1" to cflags
46 ### for openmp use -DELPH_OMP_PARALLEL_BUILD in CFLAGS and set -
       fopenmp in LD_FLAGS and CFLAGS
```

Bibliography