PW-Teleman Tutorial

author

institute

November 7, 2018

1. Useful information about PW-Teleman:

```
prompt > cd $pw-teleman_dev-all_201409/doc
prompt > ls
```

`explain-init.pdf`	installation and description of the input files
`openmp_explain.txt`	instructions for the OpenMP version of the 3D code
`modif_dev-pgr.txt`	changes in the branch 'dev-pgr'
`FFT.txt`	modified make commands
`ase-pwteleman.txt`	coupling of the PWTELEMAN code with ASE

2. Several examples that illustrate some of the features implemented in the code:

```
prompt > cd $pw-teleman_dev-all_201409/doc
```

Before you start the calculations

1. Just go in the pwteleman directory and enter `source ./install_pwteleman` and follow the instructions
2. The only prerequisite is a working `gfortran` in your path (type `sudo apt-get install gfortran` if you have root on Linux)
3. Enter `source ./install_pwteleman d` for gfortran compilation with debug options
4. `source ./install_pwteleman a` for all possible compilations, if you have Intel Fortran Compiler and NVCC
5. `source ./install_pwteleman t` to test your results and cross-correlate them against reference results (quite long)
6. At any time you can try `code/source_f90/regressionmatrix` to measure the change in your results vs reference results
7. `source ./install_pwteleman i` for GPU compilations only (if you have Intel Fortran and NVCC)
8. if it fails to `source_f90` directory :

   `cd pw-teleman_dev-all_20131009/code/source_f90/`

   Before compilation, you might want to update some settings,

Jellium model for Na8 - Real Time TDDFT

1. Go the `na8-jel` directory
   `cd pw-teleman_dev-all_20131009/samples/na8-jel`
   you should see following files:
   - `for005.na8-jel` general input for settings, static and dynamics
   - `for005` defines the qualifier na8-jel for the other `for005` files
2. Open and read the sample file `for005.na8-jel`, there are three namelists:
   - `&GLOBAL` choice of the system, initialization of wave functions, convergence issues
   - `&DYNAMIC` numerical and physical parameters for statics and dynamics, way of excitation, flags for observables
   - `&SURFACE`

In file `openmp_explain.txt` you can find complete description of the parameters used in the input.

1. Run the code using mpirun command or mpiexec depending on your installation (or include it in a batch submission):

   ```
   mpirun [mpirun_options]
   ../../code/pwteleman.par
   ```

   e.g.:

   ```
   mpirun -np 8 ../../code/pwteleman.par
   ```

   (using 8 processors to run on)

2. You will get several output files, e.g.:

`for006.0na8-jel`	main output
`energies.na8-jel`	binding energy
`pdip.na8-jel`	dipole moments
`infosp.na8-jel`	dynamical informations
`pquad.na8-jel`	quadripol moments
`penergies.na8-jel`	energy informations

Let's plot the results - Dipole moment

To plot time vs dipole moment, copy `pdip.na8-jel` into `pdip`:

```
cp pdip.na8-jel pdip
```

Open `pdip` and remove six first lines to get the readable format to gnuplot, then:

```
gnuplot> set xlabel 'Time step'
gnuplot> set ylabel 'Dipole moment'
gnuplot> unset key
gnuplot> plot './pdip' w l
```

Let’s plot the results - Dipole moment

Let's plot the results- Absorption spectra

You need to increase `ismax` and `itmax` to larger values in file `for005.na8jel` and rerun the code, maybe in batch mode To obtain the data for absorption spectra you need to compile and run a `spectr2.F90` code:

```
cd pw-teleman_dev-all_20131009/code/source_aux
gfortran spectr2.F90 -o spectr
../../code/source_aux/spectr < pdip.na8-jel >
spectra
```

Data you need to plot absorption spectra will be collected in `spectra`, make the file readable to gnuplot and then:

```
gnuplot> set xlabel 'Energy [eV]'
gnuplot> set ylabel 'oscillator strength'
gnuplot> unset key
gnuplot> set xrange [0:5]
gnuplot> plot './spectra' u 2:3 w l
```

Let's plot the results - Absorption spectra

Oscillatory motion of diatomic molecule - hydrogen

1. Go the `H2` directory

   ```
   cd pw-teleman_dev-all_20131009/samples_py/H2
   ```

   you should see following files:

`for005.H2`	general input for settings, static and dynamics
`for005`	defines the qualifier na8-jel for the other `for005` files
`for005ion.H2`	ionic configuration of cluster

2. Open and read the sample file `for005.H2`, there are three namelists:

`&GLOBAL`	choice of the system, initialization of wave functions, convergence issues
`&DYNAMIC`	numerical and physical parameters for statics and dynamics, way of excitation, flags for observables
`&PERIO`	needed when `ipsptype=1`

In file `openmp_explain.txt` you can find complete description of the parameters used in the input. You can alternatively input

Now open open sample file `for005ion.H2`, you will see two lines:

```
0.09597562751811 0.09597562746044 -0.89554425290261  1  xyz  1.0  -1
0.09597905125029 0.09597905333623  0.89609579462877  1  xyz  1.0   1
```

where:

- x,y,z coordinates
- number of element in periodic system
- only `init_lcao=1`: ordering of nodes in repeat initialization at this ion
- only `init_lcao=1`: radius of initial Gaussian at this ion
- only `init_lcao=1`: starting spin for initialization at this ion

In case when the number of the wavefunctions is too little the calculations might fail, thus, in this example we will run the code in serial.

1. Go to `source_f90` directory:

   ```
   cd ..
   cd ..
   cd code/source_f90/
   ```

2. Execute `make` command and compile the code with FFTW functions to produce serial code:

   ```
   ./make.sh 0 fftw
   ```

   The new executable called `pwteleman.seq` will be in the directory `code` which is one level below the sub-directory `source_f90`

3. Go back to the `H2` directory and run the code:

   ```
   ../../code/pwteleman.seq
   ```

1. You will get several output files, e.g.:

`for006.0H2`	main output
`energies.H2`	binding energy
`pdip.H2`	dipole moments
`infosp.H2`	dynamical informations
`pvelion.H2`	ionic velocities
`penergies.H2`	energy informations
`pposion.H2`	ionic positions

2. The last one `pposion.H2` will be needed to estimate the oscillation period of diatomic hydrogen.

3. Use plotting software to draw 'Positions vs Time' graph, e.g. with gnuplot:

```
gnuplot> set ylabel 'Positions'
gnuplot> set xlabel 'Time'
gnuplot> set xrange [0:20]
gnuplot> unset key
gnuplot> plot './pposion.H2'u 1:5 w d
```

This will show following graph:

From closer inspection you can see that the corresponding period of the oscillation is 4 units.

Coupling of the PWTELEMAN code with ASE

ASE is an Atomistic Simulation Environment written in the Python programming language with the aim of setting up, steering, and analyzing atomistic simulations.
The coupling of the PWTELEMAN code with ASE is still in progress.