PW-Teleman Tutorial

author

institute

November 15, 2013

1. Useful information about PW-Teleman:

```
prompt > cd $pw-teleman_dev-all_20131009/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_20131009/doc
```

Before you start the calculations

1. Go to `source_f90` directory :

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

   Before compilation, you might want to update some settings, for detailed explanations see `openmp_explain.txt`.

2. To use environment modules for compiling or running a code you have to load a particular module

   ```
   module load <module name>
   ```

3. Execute `make` command which you can modify in several ways (see `openmp_explain.txt`). In our example we compile the code with FFTW functions using parallelization for wavefunctions:

   ```
   ./make.sh 1 fftw
   ```

4. The executable will be copied to the working directory which is one level below the sub-directory `source_f90`. The file called `essai.par` is the new executable.

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 shell script:

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

   e.g.:

   ```
   mpirun -np 8 ../../code/essai.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

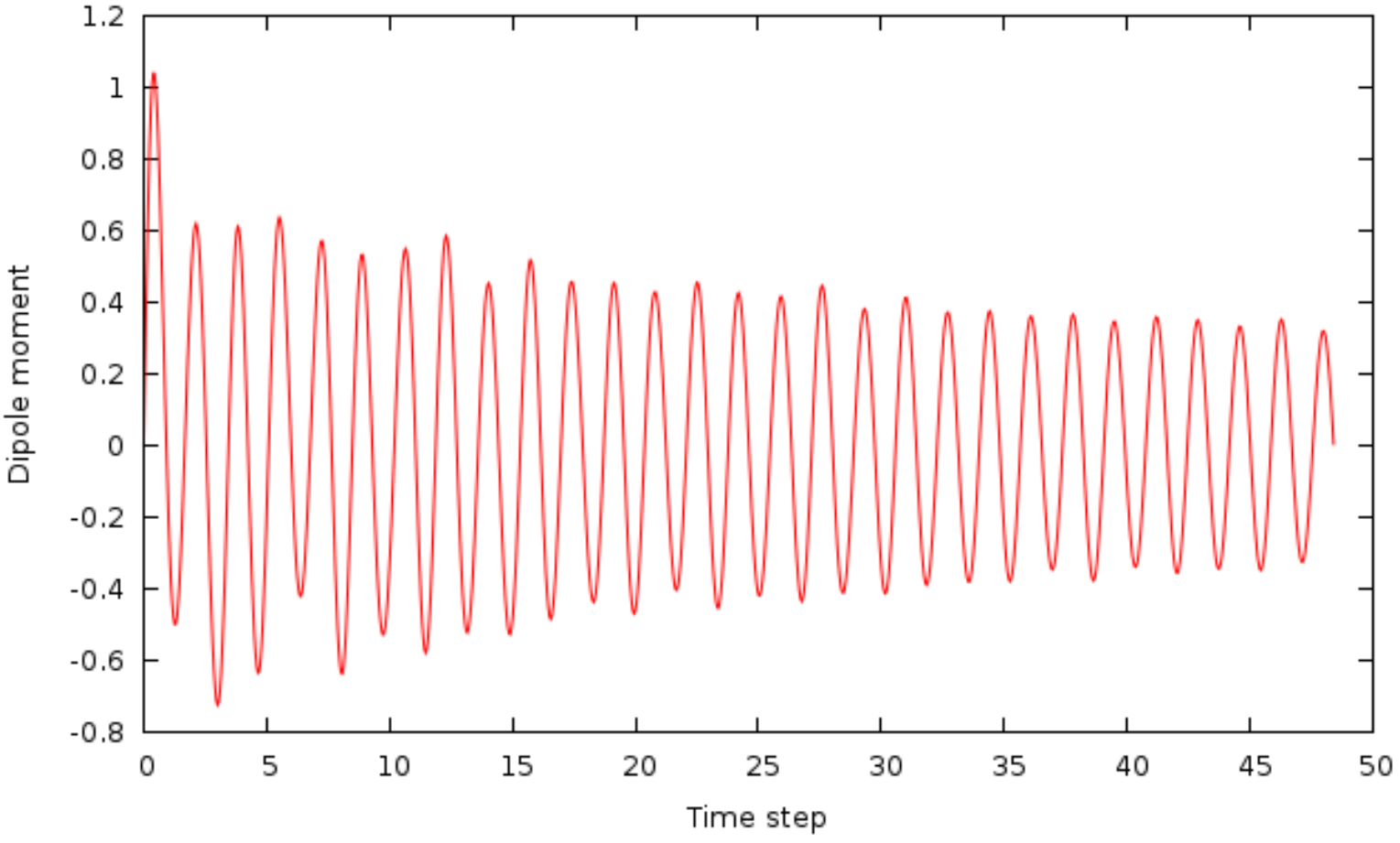
To plot time vs dipole moment, copy `pdip.na8-jel` into `pdip`:

```
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

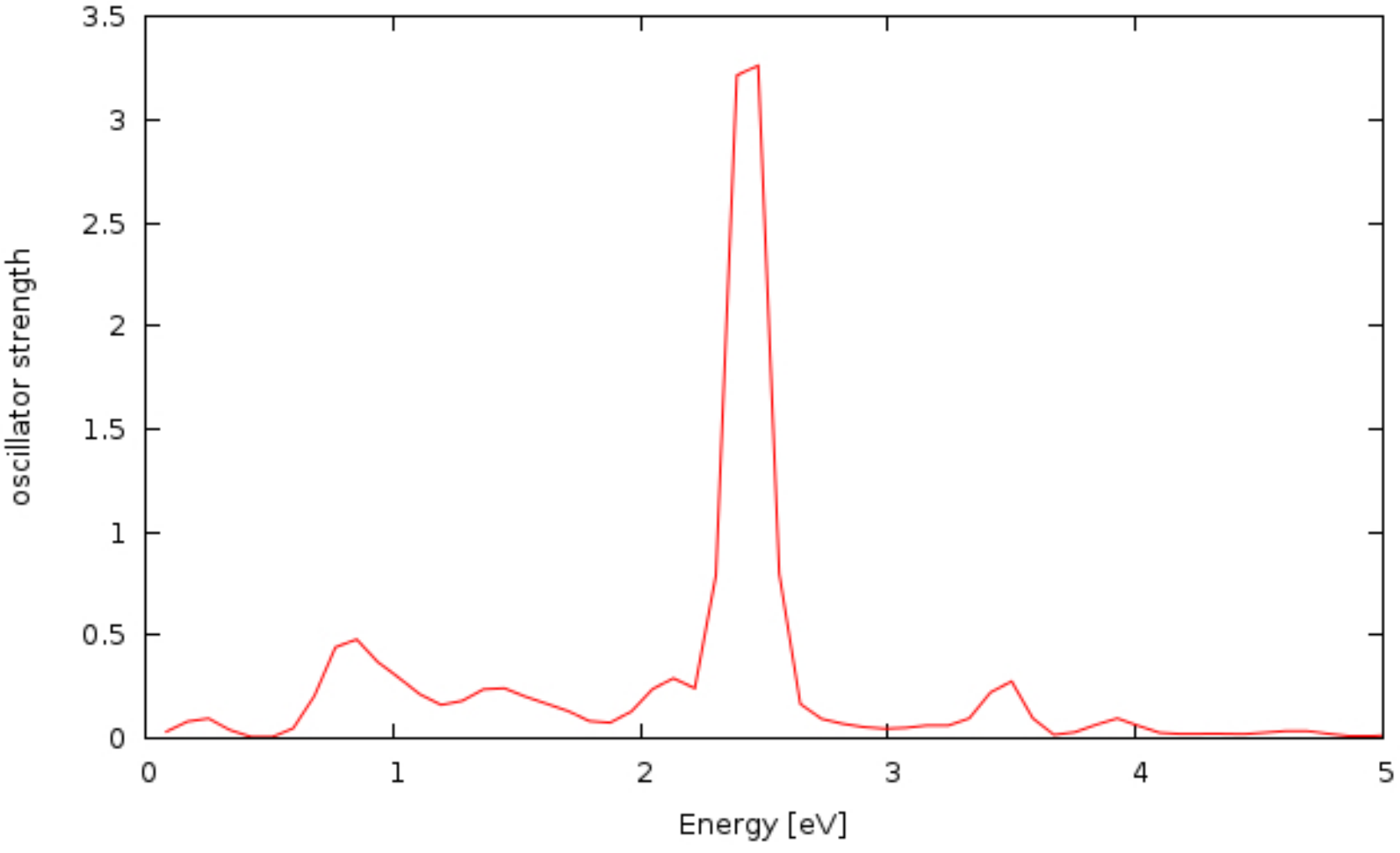
To obtain the data for absorption spectra you need to compile and run a `spectr2.F90` code:

```
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/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.

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 `essai.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/essai.seq
   ```

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

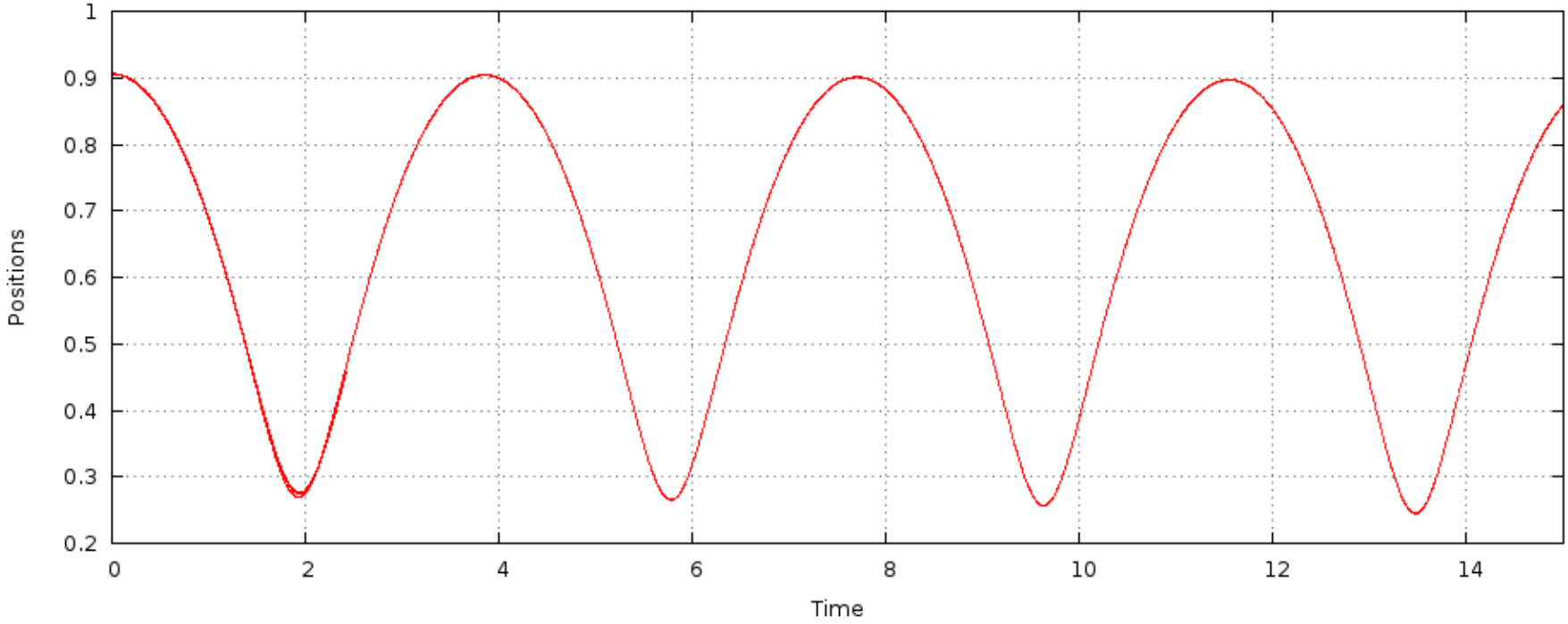
`for006.0H2`	main output
`energies.H2`	binding energy
`pdip.H2`	dipole moments
`infosp.H2`	dynamical informations
`pvelion.H2`	ionic velocities
`penergies.H2`	energy informations
`pposion.H2`	ionic positions

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 corresponds period of the oscillation is 4 ?s. TO BE CONTINUED

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.

Useful informations about ASE:

Overview click!
What is Phyton? click!
ASE and Phyton tutorials click!
Documentation click!

PWTELEMAN-ASE : An example using Na

1. First thing, you need to install ASE:
   https://wiki.fysik.dtu.dk/ase/download.html
2. Then, go to `Na` directory:
   `cd pw-teleman_dev-all_20131009/code/Python_ASE/Na/`
   You should see 3 following files :
   `na.py  pwtelemanscript.py  pwtelemandynr.py`
3. Move the file `pwtelemandynr.py` to directory `calculators`:
   `mv pwtelemandynr.py [your_ase_path]/ase/calculators/`
4. Place the file `na.py` in your working directory
5. Set `pwteleman_SCRIPT` environment variable as follow :
   `export pwteleman_SCRIPT=~/pwtelemanscript.py`
6. In the file `pwtelemanscript.py`, replace the location of PWTELEMAN code by your actual path (line 3)
7. Run the python script with the following command :
   `python na.py`

ASE will automatically generate the input files for the PWTELEMAN code (the for005* files) and then run the code itself.

Relaxation of ionic coordinates: PWTELEMAN-ASE coupling example of H2

1. Create a sub-directory `H2py` for example in `samples`:
   ```
   cd pwteleman/pw-teleman_dev-all_20131009/samples/
   mkdir H2py
   ```
2. Copy the files `na.py`,`pwtelemanscript.py` and `pwtelemandynr.py` to `H2py`
   ```
   cd H2py
   cp ../../code/Python_ASE/Na/pwtelemandynr.py .
   cp ../../code/Python_ASE/Na/pwtelemanscript.py .
   cp ../../code/Python_ASE/Na/na.py .
   ```
3. Move the file `pwtelemandynr.py` to directory `calculators`:
   ```
   mv pwtelemandynr.py [your_ase_path]/ase/calculators/
   ```
4. Set `pwteleman_SCRIPT` environment variable as follow :
   ```
   export pwteleman_SCRIPT=~/pwtelemanscript.py
   ```
5. In the file `pwtelemanscript.py`, replace the location of PWTELEMAN code by your actual path (line 3)

Now prepare input with coordinates in XYZ chemical file format. The format is as follows:

```
< of atoms>
comment line
atom1 x-coord1 y-coord1 z-coord1
atom2 x-coord2 y-coord2 z-coord2
...
atomN x-coordN y-coordN z-coordN
```

For example:

```
2

H 0.0 0.0 -1.0
H 0.0 0.0 1.0
```

Call the file `H2.xyz`

Then you should rename the file `na.py`

```
mv na.py h2.py
```

and adapt it to the needs of this example of H2 using e.g. gedit:

```
gedit h2.py
```

```
h2=read('h2.xyz')
from ase.calculators.pwteleman import pwteleman
from ase.calculators.pwtelemandynr import pwtelemandynr

h2.set_calculator(pwtelemandynr(isurf=0,dx=0.2354,dy=0.2354,
dz=0.2354,
kxbox=24,kybox=24,kzbox=48,nspdw=1,init_lcao=1,ismax=20,
modecalc='static'))
e = h2.get_potential_energy()
print e
traj=PickleTrajectory('h2.traj','w')
dyn=QuasiNewton(h2).run(fmax=0.0001)
e = h2.get_potential_energy()
print e-4*13.6
```

The last step is to run a calculations:

```
python h2.py
```

All for005* files will be generated automatically by ASE and then the code will run itself.
In file `for005ion.pwteleman` you can find the final coordinates after relaxation.