ABINIT, lesson on the temperature-dependence of the electronic structure (TD)

This lesson aims at showing how to get the following physical properties, for periodic solids:

- The zero-point-motion renormalization (ZPR) of eigenenergies
- The temperature-dependence of eigenenergies
- The lifetime/broadening of eigenenergies

This lesson should take about 1 hour.

Copyright (C) 2005-2014 ABINIT group (XG)
This file is distributed under the terms of the GNU General Public License, see ~abinit/COPYING or http://www.gnu.org/copyleft/gpl.txt.
For the initials of contributors, see ~abinit/doc/developers/contributors.txt.

General note

At the time of writing (June 2014), there are two ways to compute the temperature dependence with Abinit:

- **Using Anaddb**: historically the first coded way. This option does not require the use of Netcdf
- **Using post-processing python scripts**: this is the recommended way as it provide more options and is more optimal (take less disk space and is less memory demanding). This option REQUIRED the use of Netcdf (both for Abinit and for python). To install Abinit with Netcdf please see the following link (SP: TO BE ADDED). You should also install netcdf for python. Informations can be found here.

For the theory related to the temperature dependence calculations, you can read the following papers:


This tutorial can only be done with Abinit 7.9.1 and higher.

1. Calculation of the ZPR of eigenenergies at $\mathbf{q} = \mathbf{\Gamma}$

All the inputs files are located in ~/abinit/tests/tutorespfn/Input/temp_x and the reference files in ~/abinit/tests/tutorespfn/Refs/temp_x but we will provide here all the input files that you can
copy/paste to any place that suits you to do the calculations.

First, run the calculation using the following temp_1.in input file

```plaintext
# C in diamond structure.
ndtset 3
elph2_imagden 0.1 eV   # Imaginary shift of the denominator of the sum-over-states
# in the perturbation denominator. Usual value is 0.1 eV to reproduce
# experimental broadening at 300K. Increasing
# convergence with respect to the number of q-points.
ngkpt 2 2 2            # k-grid should be at least 4x4x4 for diamond to be converged.
nshiftk 1
shiftk 0.0 0.0 0.0

# Ground state density
iscf1    7
tolvrs1  1.0d-8      # tolvrs 1.0d-18 should be used for converged results

# Non self-consistent calculation with an arbitrary q point (here Gamma)
getden2  1
iscf2   -2
nqpt2   1
qpt2    0.0 0.0 0.0
nbdbuf2  2
tolwfr2 1.0d-8       # tolwfr 1.0d-22 should be used for converged results

# Computation at Gamma
getwfk3  1
getwfq3  2
nqpt3   1
qpt3    0.0 0.0 0.0
ieig2rf3 1            # Static eigenvalues corrections using DFPT (Sternheimer)
smdelta3 1            # Flag required to produce the _EIGI2D used to compute the lifetime of electronic levels.
# smdelta = 1 ==> Fermi-Dirac smearing.
bdbuf3 2            # 2 buffer bands. RF converges much faster.
rfphon3 1            # Do phonon response
rfatpol3 1 2          # Treat displacements of all atoms
rfdir3  1 1 1          # Do all directions
tolwfr3 1.0d-8        # tolwfr 1.0d-22 should be used for converged results
```

You can use the following files file for abinit (will always be the same, just change the last digit for the other calculations):

```
temp_1.files
```

```
temp_1.in
temp_1.out
temp_1i
temp_1o
temp_1
../.../../Pspfs_for_tests/6c.pspnc
```

**If Abinit is compiled with Netcdf**

If you have compiled Abinit with Netcdf, the calculation will produce _EIG.nc, _DDB, EIGR2D.nc and EIGI2D.nc that contain respectively the eigenvalues (GS or perturbed), the second-order derivative of the total energy with respect to two atomic displacement, the electron-phonon matrix elements used to compute the renormalization of the eigenenergies and the electron-phonon matrix elements used to compute the lifetime of the electronic states.

You can then copy the post-processing (PP-temperature) python file from ~/abinit/scripts/post_processing/temperature_para.py as well as the python file containing the required classes from ~/abinit/scripts/post_processing/mrgeignc.py into your current directory where you did the Abinit calculations.

You can then simply run the python script with the following command

```
python temperature_para.py
```
and enter the informations that the script ask.

For example a set of possible answer could be (please remove the comments if you plan to use this file as input for the script)

<table>
<thead>
<tr>
<th>Number of cpu to do the calculations</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td># Static ZPR computed in the Allen-Heine-Cardona (AHC) theory</td>
<td></td>
</tr>
<tr>
<td># Root name of the output</td>
<td></td>
</tr>
<tr>
<td># We want the ZPR AND the temperature dependence</td>
<td></td>
</tr>
<tr>
<td># We want the renormalization between 0 and 1000K by steps of 50K.</td>
<td></td>
</tr>
<tr>
<td># Number of Q-points we have (here we only computed</td>
<td>1</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td></td>
</tr>
<tr>
<td># Name of the response-funtion (RF) DDB file</td>
<td></td>
</tr>
<tr>
<td># Eigenvalues at $\mathbf{k+q}$</td>
<td></td>
</tr>
<tr>
<td># Second-order electron-phonon matrix element</td>
<td></td>
</tr>
<tr>
<td># Second-order electron-phonon matrix element for electronic lifetime</td>
<td></td>
</tr>
<tr>
<td># Eigenvalues at $k$</td>
<td></td>
</tr>
</tbody>
</table>

The python code will generate 3 files:

- **temperature.txt**: This text file contain the ZPM correction at each \$\mathbf{k}\$-points for each bands. It also contain the evolution of each band with temperature at \$\mathbf{k}=$\Gamma\$. At the end of the file, the Fan/DDW contribution is also reported.

- **temperature_EP.nc** This netcdf file contain a number for each \$\mathbf{k}\$-points, for each bands and each temperature. The real part of this number is the ZPM correction and the imaginary part is the lifetime.

- **temperature_BRD.txt** This text file contain the lifetime of the electronic states at each \$\mathbf{k}\$-points for each bands. It also contain the evolution of each band with temperature at \$\mathbf{k}=$\Gamma\$.

We can for example visualize the temperature dependence at \$\mathbf{k}=$\Gamma\$ of the HOMO bands with the contribution of only the \$\mathbf{q}=$\Gamma\$ from the **temperature.txt** file.

⚠️ The data plotter plugin is still experimental. Please check that everything is OK. *Yann Pouillon, 2014/08/25*
Here you can see that the HOMO correction goes down with temperature. This is due to the highly non-converged calculation. If you increase $\text{ecut}$ from 5 to 10, you get the following plot.

This means that the HOMO eigenenergies correction will go up with temperature. You can also plot the LUMO eigenenergies corrections and see that they go down. The ZPR correction as well as their temperature dependence usually closes the gap in semiconductors.

**If Abinit is not compiled with Netcdf**

Here we should first use $\text{mrgddb}$ to merge the _DDB and _EIGR2D/_EIGI2D but since we only have one $\mathbf{q}$-point we do not have to do it.

The static temperature dependence as well as the G2F can be computed thanks to $\text{anaddb}$ with the following files file:

```
temp_2.files
    temp_2.in
    temp_2.out
    temp_1o_DS3_DDB
dummyo.md
temp_1o_DS3_EIGR2D
temp_2
dummy.ddk
```

and the following input file:

```
temp_2.in

!Input file for the anaddb code. Analysis of the C DDB
```

Tips for ABINIT users and developers - https://wiki.abinit.org/
The run will create three files:

- **temp.2.out.ep_G2F**: This $g^2F$ spectral function represent the contribution of the phononic modes of energy $E$ to the change of electronic eigenenergies according to the following equation

$$g^2F(n\mathbf{k},E)=\sum_{\mathbf{q}m}\frac{\partial \varepsilon_{nm}(\mathbf{k})}{\partial \varepsilon_{nm}(\mathbf{q})}\delta(E-\hbar\omega_m(\mathbf{q}))$$

- **temp.2.out.ep_PDS**: This file contains the phonon density of states
- **temp.2.out.ep_TBS**: This file contains the eigenenergies corrections as well as the temperature dependence one. You can check that the results are the same as with the python script approach here above.

**2. Converging the calculation with the \textbf{$q$}-point grid**

Starting now we are going to only describe the approach with Abinit compiled with Netcdf. The approach with Anaddb is similar to what we described above. Note that Anaddb only support homogenous $\mathbf{q}$-point integration.
You can integrate over the $\mathbf{q}$-grid using either random $\mathbf{q}$-point integration or homogenous Monkhorst-Pack based integration. For the random integration you should create a script that generate random $\mathbf{q}$-point, perform the Abinit calculations at these points and then integrate them using the temperature_para.py script. The script will detect that you did random integration thanks to the weight of the $\mathbf{q}$-point stored in the _EIGR2D.nc file and perform the integration accordingly. The random integration converges slowly but in a consistent manner. Since this methods is a little bit less user friendly we will focus on the homogenous integration.

The first thing we need to do is to determine the number of $\mathbf{q}$-point in the IBZ for a given $\mathbf{q}$-point grid. We choose here a 4x4x4 $\mathbf{q}$-point grid.

Using the following input file

```
# C in diamond structure.
ndtset 3 udtset 1 3
iqpt:? 1               # Index of the first q-point of this file
(usefull if you split your
# input files

elph2_imagden 0.1 eV  # Imaginary shift of the denominator of the
sum-over-states # in the perturbation denominator. Usual value
is 0.1 eV to reproduce # experimental broadening at 300K. Increasing
the value help the # convergence with respect to the number of q-
points.
ngkpt 2 2 2            # k-grid should be at least 4x4x4 for diamond
to be converged.
nshiftk 1
shiftk 0.0 0.0 0.0
ngqpt 4 4 4            # Should be converged upon
qptopt 1
nshiftq 1
shiftq 0.0 0.0 0.0

# Ground state density
iqpt+?    1
iscf?1    7
tolvrs?1  1.0d-8   # tolvrs 1.0d-18 should be used for converged
results

# Non self-consistent calculation with an arbitrary q point (here Gamma)
getden?2 -1
iscf?2   -2
ngpt?2   1
nbdbuf?2  2
tolwfr?2 1.0d-8   # tolwfr 1.0d-22 should be used for converged
```

Tips for ABINIT users and developers - https://wiki.abinit.org/
results

# Computation at Gamma
getwfk?3 -2
getwfq?3 -1
nqpt?3 1
ieig2rf?3 1 # Static eigenvalues corrections using DFPT (Sternheimer)
smdelta?3 1 # Flag required to produce the _EIGI2D used to # compute the lifetime of electronic levels.
 # smdelta = 1 ==> Fermi-Dirac smearing.
nbdbuf?3 2 # 2 buffer bands. RF converges much faster.
rfphon?3 1 # Do phonon response
rfatpol?3 1 2 # Treat displacements of all atoms
rfdir?3 1 1 1 # Do all directions
tolwfr?3 1.0d-8 # tolwfr 1.0d-22 should be used for converged results

# Cell dependant parameters
acell 3*6.675
rprim 0 .5 .5 0 .5 .5 .5 0
nsym 1 # Symmetries are not yet sufficiently tested.
 # Disable symmetries.
natom 2
typat 1 1
xred 3*0.0 3*0.25
nband 12
ntypat 1
znucl 6
diemac 6
ecut 5 # Underconverged ecut.
enunit 2
nsstep 50
istwfk *1

you can launch it and quickly kill the job. Then look into the log file to find the following line after the list of $\mathbf{q}$-points:

```
symkpt : the number of k-points, thanks to the symmetries, is reduced to 8
```

In general, in order to get the number of $\mathbf{k}$-points, you launch a “fake” run with a $\mathbf{k}$-point grid equivalent to the $\mathbf{q}$-point grid you want to use in your calculation.

Now that we know that the 4x4x4 $\mathbf{q}$-point grid reduces to 8 IBZ $\mathbf{q}$-point we can make the following substitution into the input file

```
ndtset 3 udtset 1 3 ==> ndtset 24 udtset 8 3
```
and then launch the calculation. When the Abinit run is finished you can launch the python script

```python
python temperature_para.py < temperature3.files
```

with the following file

```python
temperature3.files

2
1
  temperature3
y
1000 50
y
8
temp_3o_DS13_DDB
temp_3o_DS23_DDB
temp_3o_DS33_DDB
temp_3o_DS43_DDB
temp_3o_DS53_DDB
temp_3o_DS63_DDB
temp_3o_DS73_DDB
temp_3o_DS83_DDB
temp_3o_DS12_EIG.nc
temp_3o_DS22_EIG.nc
temp_3o_DS32_EIG.nc
temp_3o_DS42_EIG.nc
temp_3o_DS52_EIG.nc
temp_3o_DS62_EIG.nc
temp_3o_DS72_EIG.nc
temp_3o_DS82_EIG.nc
temp_3o_DS13_EIGR2D.nc
temp_3o_DS23_EIGR2D.nc
temp_3o_DS33_EIGR2D.nc
temp_3o_DS43_EIGR2D.nc
temp_3o_DS53_EIGR2D.nc
temp_3o_DS63_EIGR2D.nc
temp_3o_DS73_EIGR2D.nc
temp_3o_DS83_EIGR2D.nc
```

The plotting of the same HOMO band at $\mathbf{k} = \mathbf{\Gamma}$ for a 4x4x4 $\mathbf{q}$-
point grid gives a very different result than previously (this graph has been obtained with `ecut = 10`).

As a matter of fact, diamond needs an extremely dense $\mathbf{q}$-point grid (40x40x40) to be converged. On the bright side each $\mathbf{q}$-point calculation is independent and thus scales ideally...

### 3. Calculation of the eigenenergies correction along high-symmetry lines

The calculation of the electronic eigenvalue correction due to electron-phonon coupling along high-symmetry lines (also known as electronic bandstructure) require the use of 6 datasets per $\mathbf{q}$-points. Different datasets are required to compute the following quantities:

1. $\Psi_{\mathbf{k}_{\text{HS}}}^{(0)}$ the ground-state wavefunction on the Homogeneous $\mathbf{q}$-point sampling.
2. $\Psi_{\mathbf{k}_{\text{BS}}}^{(0)}$ the ground-state wavefunction computed along the bandstructure $\mathbf{q}$-point sampling.
3. $\Psi_{\mathbf{k}_{\text{HS}}+\mathbf{q}}^{(0)}$ the ground-state wavefunction on the shifted Homogeneous $\mathbf{q}$-point sampling.
4. $\Psi_{\mathbf{k}_{\text{BS}}+\mathbf{q}}^{(0)}$ the ground-state wavefunction obtained from reading the perturbed density of the previous dataset
5. Reading the previous quantity we obtain the el-ph matrix elements along the BS with all physical quantities integrated over a HS grid.

**temp_4.in**

```plaintext
# C in diamond structure.
ndset 48 udtset 8 6
iqpt:? 1               # Index of the first q-point of this file
(usefull if you split your # input files
 # input files

elph2_imagden 0.1 eV   # Imaginary shift of the denominator of the
sum-over-states        # in the perturbation denominator. Usual value
is 0.1 eV to reproduce # experimental broadening at 300K. Increasing
the value help the    # convergence with respect to the number of q-
```

https://wiki.abinit.org/  Printed on 2019/06/14 14:09
points.
gnkpt 2 2 2            # k-grid should be at least 4x4x4 for diamond  
to be converged.
nshiftk 1
shiftk 0.0 0.0 0.0
ngqpt 4 4 4            # Should be converged upon
qptopt 1
nshiftq 1
shiftq 0.0 0.0 0.0

# Ground state density
iqpt+?  1
iscf?1  7
tolvrs?1 1.0d-8     # tolvrs 1.0d-18 should be used for converged results

# Non self-consistent calculation following high sym k path
getden?2  -1
iscf?2   -2
getwfk?2  -1
tolwfr?2 1.0d-8     # tolwfr 1.0d-22 should be used for converged results
kptopt?2  -9
ndivsm?2  5
kptbounds?2
  1/2 0.0 0.0 0.0  # L
  0.0 0.0 0.0 0.0  # Gamma
  0.0 1/2 1/2 1/2  # X
  1/4 1/2 3/4 1/2  # W
  3/8 3/8 3/4 1/2  # K
  1/2 1/2 1/2 1/2  # L
  1/4 1/2 3/4 1/2  # W
  1/2 1/2 1.0 1/2  # X
  3/8 3/8 3/4 1/2  # K
  0.0 0.0 0.0 0.0  # Gamma

# Non self-consistent calculation with an abritrary q point
getden?3 -2
getwfk?3 -2
iscf?3  -2
nqpt?3   1
nbdbuf?3 2
tolwfr?3 1.0d-8     # tolwfr 1.0d-22 should be used for converged results

# Computation at q
getwfk?4  -3
getwfgq?4 -1
iscf?4    7
nqpt?4    1
ieig2rf?4 1            # Static eigenvalues corrections using DFPT
(Sternheimer)

smdelta?4 1
# Flag required to produce the _EIGI2D used to
# compute the lifetime of electronic levels.
# smdelta = 1 ==> Fermi-Dirac smearing.
nbdbuf?4 2
# 2 buffer bands. RF converges much faster.
rphon?4 1
# Do phonon response
rfatpol?4 1 2
# Treat displacements of all atoms
rfdim?4 1 1 1
# Do all directions
tolwfr?4 1.0d-8
# tolwfr 1.0d-22 should be used for converged
# results

# Non self-consistent calculation following high sym k path
nqpt?5 1
gdden?5 -4
iscf?5 -2
getwfk?5 -3
nbdbuf?5 2
# 2 buffer bands. RF converges much faster.
tolwfr?5 1.0d-8
# tolwfr 1.0d-22 should be used for converged
# results
kptopt?5 -9
ndivsm?5 5
kptbounds?5
  1/2 0.0 0.0  # L
  0.0 0.0 0.0  # Gamma
  0.0 1/2 1/2  # X
  1/4 1/2 3/4  # W
  3/8 3/8 3/4  # K
  1/2 1/2 1/2  # L
  1/4 1/2 3/4  # W
  1/2 1/2 1.0  # X
  3/8 3/8 3/4  # K
  0.0 0.0 0.0  # Gamma

# Computation at an other q point
nqpt?6 1
gdden?6 -5
get1den?6 -2
getwfk?6 -4
getwfaq?6 -1
tolwfr?6 1.0d-8
# tolwfr 1.0d-22 should be used for converged
# results
nbdbuf?6 2
# 2 buffer bands. RF converges much faster.
ieig2rf?6 1
smdelta?6 1
rphon?6 1
rfatpol?6 1 2
rfdim?6 1 1 1
iscf?6 -2
kptopt?6 -9
ndivsm?6 5
Using the following files file for the python script

```python
# temperature4.files

1
1
temperature4
y
1000 50
y
8
temp_4o_DS14_DDB
temp_4o_DS24_DDB
temp_4o_DS34_DDB
temp_4o_DS44_DDB
temp_4o_DS54_DDB
temp_4o_DS64_DDB
temp_4o_DS74_DDB
temp_4o_DS84_DDB
temp_4o_DS15_EIG.nc
temp_4o_DS25_EIG.nc
```
temp_4o_DS35_EIG.nc
temp_4o_DS45_EIG.nc
temp_4o_DS55_EIG.nc
temp_4o_DS65_EIG.nc
temp_4o_DS75_EIG.nc
temp_4o_DS85_EIG.nc
temp_4o_DS16_EIGR2D.nc
temp_4o_DS26_EIGR2D.nc
temp_4o_DS36_EIGR2D.nc
temp_4o_DS46_EIGR2D.nc
temp_4o_DS56_EIGR2D.nc
temp_4o_DS66_EIGR2D.nc
temp_4o_DS76_EIGR2D.nc
temp_4o_DS86_EIGR2D.nc
... (list continues with similar files)

Off course the high symmetry points that we computed in section 2 have the same value here. It is a good idea to check it by running the following script file

temperature3_bis.files

1
1
temperature4_bis
y
1000 50
y
8
temp_4o_DS14_DDB
temp_4o_DS24_DDB
temp_4o_DS34_DDB
temp_4o_DS44_DDB
temp_4o_DS54_DDB
temp_4o_DS64_DDB
temp_4o_DS74_DDB
temp_4o_DS84_DDB
temp_4o_DS13_EIG.nc
temp_4o_DS23_EIG.nc
temp_4o_DS33_EIG.nc
temp_4o_DS43_EIG.nc
temp_4o_DS53_EIG.nc
temp_4o_DS63_EIG.nc
You can then copy the plotting script (Plot-EP-BS) python file from
~/abinit/scripts/post_processing/plot_bs.py into your current directory where you did the Abinit calculations. Note that in order to run this script you need the following requirement

- python 2.7.6 or higher
- numpy 1.7.1 or higher
- netCDF4 and netCDF4 for python
- scipy 0.12.0 or higher

You can then use the following script feeding file or enter the data manually

```
plot_bs.files
```

```
temperature4_EP.nc
L \Gamma X W K L W X K \Gamma
-20 30
0
```

This should give you the following bandstructure:
where the solid black lines are the traditional electronic bandstructure, the dashed lines are the electronic eigenenergies with electron-phonon renormalization at a defined temperature (here 0K). Finally the area around the dashed line is the lifetime of the electronic eigenstates.

You should of course notice all the spiky jumps in the renormalization. This is due to the fact that we did a completely under-converged calculation.

It is possible to converge the calculations using an \texttt{ecut} of 30 Ha, a \texttt{ngkpt} grid of 6x6x6 and an increasing \texttt{ngqpt} grid to get converged results:
As you can see the limiting factor for the convergence study is the convergence of the LUMO band at $k=\Gamma$. This band is not the lowest in energy (the lowest in on the line between $\Gamma$ and $X$) and therefore this band is rather unstable. This can also be seen by the fact that it has a large electronic broadening, meaning that this state will decay quickly into another state.

Using the relatively dense $\mathbf{q}$-grid of $43\times43\times43$ we can obtain the following converged bandstructure:

<table>
<thead>
<tr>
<th>q-grid</th>
<th>Nb qpt in IBZ</th>
<th>$k=\Gamma_{25'}$</th>
<th>Broadening</th>
<th>$k=\Gamma_{15}$</th>
<th>Lifetime</th>
<th>$k=(\Gamma-X)$</th>
<th>Broadening</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4x4</td>
<td>8</td>
<td>0.1175</td>
<td>0.0701</td>
<td><strong>-0.3178</strong></td>
<td>0.1916</td>
<td>-0.1570</td>
<td>0.0250</td>
</tr>
<tr>
<td>10x10x10</td>
<td>47</td>
<td>0.1390</td>
<td>0.0580</td>
<td><strong>-0.3288</strong></td>
<td>0.1847</td>
<td>-0.1605</td>
<td>0.0308</td>
</tr>
<tr>
<td>20x20x20</td>
<td>256</td>
<td>0.1446</td>
<td>0.0574</td>
<td><strong>-0.2691</strong></td>
<td>0.1823</td>
<td>-0.1592</td>
<td>0.0298</td>
</tr>
<tr>
<td>26x26x26</td>
<td>511</td>
<td>0.1448</td>
<td>0.0573</td>
<td><strong>-0.2736</strong></td>
<td>0.1823</td>
<td>-0.1592</td>
<td>0.0297</td>
</tr>
<tr>
<td>34x34x34</td>
<td>1059</td>
<td>0.1446</td>
<td>0.0573</td>
<td><strong>-0.2699</strong></td>
<td>0.1821</td>
<td>-0.1591</td>
<td>0.0297</td>
</tr>
<tr>
<td>43x43x43</td>
<td>2024</td>
<td>0.1447</td>
<td>0.0572</td>
<td><strong>-0.2650</strong></td>
<td>0.1821</td>
<td>-0.1592</td>
<td>0.0297</td>
</tr>
</tbody>
</table>
Here we show the renormalization at a very high temperature of 1900K in order to highlight more the broadening and renormalization that occurs. If you want accurate values of the ZPR at 0K you can look at the table above.

Possible issue while converging your calculations

If you use a extremely fine $\mathbf{q}$-point grid, the acoustic phonon frequencies for $\mathbf{q}$-points close to $\Gamma$ will be wrongly determined by Abinit. Indeed in order to have correct phonon frequencies, you have to impose the acousting sum rule with anaddd. Since it is not possible here with the script, we reject the contribution of the acoustic phonon close to $\Gamma$ if their phonon frequency is lower than $1E-6$ Ha. Otherwise you get unphysically large
ABINIT, lesson on the temperature-dependence of the electronic structure (TD)

contribution. You can tune this parameter by editing the variable “tol6 = 1E-6” in the beginning of the mrgeignc.py file. For example, for the last 43x43x43 calculation, I had to set it to 1E-4.

From: https://wiki.abinit.org/ - Tips for ABINIT users and developers

Permanent link: https://wiki.abinit.org/doku.php?id=tutorials:temperature

Last update: 2016/04/24 15:09