

A Rough Guide to ELNES/EELS with Castep

Phil Hasnip

March 27, 2008

1 Introduction

Castep 4.3 is the first version of Castep to be capable of generating electron energy loss spectra (EELS); specifically, it can calculate the near-edge spectrum (ELNES). It has been developed by Shangpeng Gao, formerly of the TCM Group in the Cavendish Laboratory, Cambridge (UK) and now based at Fudan University, Shanghai (China).

The raw calculated spectrum is broadened according to a set of input parameters, and both the raw and broadened spectra are written out.

2 Castep

Castep is a plane-wave pseudopotential density functional program designed to calculate the groundstate energy and charge density of a periodic system of electrons and ions.

When you run Castep you need to give it a seedname as an argument – Castep uses this seedname as a prefix for both the input files it expects and its output files. The main input files are:

- `<seedname>.cell`
The cell file defines the simulation cell. It contains the lattice vectors of the simulation cell, the type and positions of the atoms, the desired k-point sampling and the pseudopotential data, amongst other things.
- `<seedname>.param`
The param file specifies the parameters Castep uses for the calculation. Amongst the more common are the plane-wave cut-off energy, the extra conduction bands to include in the calculation, whether to optimise for maximum speed or minimum memory. To specify an ELNES calculation use `task: ELNES`.

Castep writes a number of output files. Some of these output files are always written, and some are task-specific. The main output files are:

- `<seedname>.castep`
This is the main Castep output file. It is human-readable, and contains a summary of the current calculation (both inputs and results).
- `<seedname>.check`
The check file contains all the data needed for checkpointing a Castep calculation, including a copy of the simulation cell data, the parameters used, the Kohn-Sham wavefunctions and the valence charge density. Castep calculations can be restarted from a check file.
- `<seedname>.castep_bin`
This file is like a check file, but does not contain the Kohn-Sham wavefunctions. It can be used to restart a Castep calculation, but Castep will have to regenerate the Kohn-Sham wavefunctions first.
- `<seedname>.paw`
This file is only generated by an ELNES calculation, and contains the unbroadened spectrum, corrected for the PAW terms.
- `<seedname>.elnes`
This file is only generated by an ELNES calculation, and contains the broadened spectrum.
- `<seedname>.<node_id>.err`
This file contains any error messages from the Castep run. Empty error files are deleted at the end of the calculation.

2.1 The Cell file

The cell file defines the simulation cell and related data. It is a freeform text file, so the order of the data is irrelevant. It is not case sensitive. Many of the data entries are contained inside “blocks”; these regions are defined by “%block `<block_name>`” at the beginning, and “%endblock `<block_name>`”, e.g.

```
%block POSITIONS_FRAC
  C 0.000000000 0.000000000 0.000000000
  C 0.250000000 0.250000000 0.250000000
%endblock POSITIONS_FRAC
```

2.1.1 Lattice Vectors

These are specified in a block although you have a choice, either ABC:

```
%block lattice_abc
<length unit>
<a> <b> <c>
<alpha> <beta> <gamma>
%endblock lattice_abc
```

or explicit lattice vectors:

```
%block LATTICE_CART
<length unit>
<row vector a>
<row vector b>
<row vector c>
%endblock LATTICE_CART
```

In each case the length unit is optional, the default being Angstroms. See appendix A for details of Castep units and related parameters.

2.1.2 Atomic coordinates

These are specified in a block, either `positions_frac` for fractional positions, or `positions_abs` for absolute Cartesian positions. In each case the basic format is the same:

```
%block POSITIONS_ABS
<length unit>
<species symbol> <x-coord> <y-coord> <z-coord>
%endblock POSITIONS_ABS
```

The length unit is optional, and the default is Angstroms; naturally this is not necessary for fractional coordinates.

The species symbol is the usual 1-, 2- or 3-letter abbreviation for the chemical element. Optionally you may choose to label these, and this is done by appending the symbol name with a colon and a tag of your choice, e.g. “C:exc”; however, the total length of the species symbol must be no more than 8 characters.

2.1.3 Brillouin Zone Sampling (k-points)

Because the k-point sampling may be different for different tasks, there are a variety of k-point related keywords and blocks. The k-points for the main groundstate calculation (a “single-point energy” calculation in Castep parlance) may be specified as:

```
kpoint_mp_spacing <reciprocal length> <reciprocal length unit>
```

where Castep will generate a Monkhorst-Pack mesh whose k-points are no more than the specified distance from their nearest neighbours. The reciprocal length unit is optional, and defaults to inverse angstroms.

Alternatively the grid dimensions may be specified directly:

```
kpoint_mp_grid <no. in x> <no. in y> <no. in z>
```

or an explicit list of k-points may be given

```

%block kpoint_list
<x-coordinate> <y-coordinate> <z-coordinate> <weight>
...
%endblock kpoints_list

```

When running ELNES calculations, Castep takes the k-points from the *band-structure* k-points, which have the same names as the ones for the ground-state calculation but prefixed by ‘bs_’, i.e. `bs_kpoint_mp_spacing`, `bs_kpoint_mp_grid` and `%block bs_kpoint_mp_list`. In fact for pure band-structure calculations you can also specify a k-point path, but this is of little use for ELNES.

2.1.4 Pseudopotentials

The pseudopotential data is contained in the block `species_pot`, and may contain either a path and filename to a pseudopotential file or a pseudopotential definition string. Each line defines a single ionic species.

Castep can read pseudopotential files from the Accelrys database, Vanderbilt’s pseudopotential generator, OPIUM, Troullier-Martin’s code. The path given may be absolute or relative to the current working directory, e.g.

```

%block species_pot
Si:std /usr/local/castep/pspot/Si_00.usp
Si:nc Si_00.recpot
...
%endblock species_pot

```

However most pseudopotential file formats do not contain enough information for the PAW reconstruction needed for ELNES, so it is not possible to use library pseudopotential files for any atom you wish to include in the ELNES calculation.

Castep can generate ultrasoft pseudopotentials on-the-fly, which will store all of the information required for the PAW reconstruction; to control this you need to put a pseudopotential definition string in place of the filename in a `species_pot` block, e.g.:

```

%block species_pot
C:otf 2|1.4|1.4|1.3|6|10|12|20:21(qc=6)
...
%endblock species_pot

```

The first integer is the angular momentum quantum number of the local channel (2 in the case above, so it’s the d channel). At the time of writing (Castep 4.3) angular momenta up to and including f (l=3) are supported.

The next three numbers are the core radii for the local-, non-local and augmentation charges respectively (in atomic units, i.e. Bohr). The next three integers are the suggested cut-off energies for “coarse”, “medium” and “fine” precision with this pseudopotential – clearly this can only come through testing

by the user, but as it doesn't affect the actual pseudopotential in any way it can be ignored initially.

The next section contains the electrons to be treated as valence, specified by the principal and orbital angular momentum numbers n and l , i.e. 20:21 means treat 2s and 2p electrons as valence; Castep will pseudise everything else.

There are several optional elements, which are best illustrated with examples:

```
%block species_pot
  H  1|0.8|0.8|0.6|2|6|8|10(qc=6)
%endblock species_pot
```

The 'qc=6' in round brackets specifies the qc-tuning for the pseudopotential.

```
%block species_pot
  Mg 2|1.6|2|1.4|6|7|8|30NH:21U:31UU:32LGG(qc=4.5) []
%endblock species_pot
```

'N' generates norm-conserving beta-projectors for that state, 'U' generates ultrasoft beta-projectors (the default); similarly 'H' generates norm-conserving gamma-projectors for that state, and 'G' generates ultrasoft gamma-projectors. Subsequent characters for the same state specify additional projectors, so 'UU' means generate two beta-projectors for this state; the default it to use two projectors per state (NB the number of gamma-projectors is always at least the number of beta-projectors).

'L' signifies that this state is in the local channel. Clearly the angular momentum number of any state with 'L' specified must match the first integer of the string.

```
%block species_pot
  Hg 1|2.2|2.2|1.7|8|10|11|60:61P:52
%endblock species_pot
```

'P' means don't use any projectors for this state, just pseudise.

```
%block species_pot
Lr 2|2.1|2.0|1.4|15|17|18|60U:70U:61:53:62P(qc=6.5){6d0}
C:exc 2|1.4|1.4|1.3|6|10|12|20:21{1s1,2s2,2p3}(qc=6) [] ! add core hole
%endblock species_pot
```

Sometimes you might want to generate a pseudopotential where the electrons do not fill up the states in order of lowest n, l first. This might be because the states have an unusual ordering in energy (e.g. Lr above) or because we want to generate a core-excited pseudopotential (e.g. C:exc above). Simply place the atomic test configuration in curly brackets {...} as in the above examples.

```
%block species_pot
  Mn 3|2.3|2.3|1.5|6|8|9|40N:32:41N(qc=4.5,q0=3){4s0.75,4p0.25}
%endblock species_pot
```

The `qc`-tuning can be altered for each angular momentum channel using `q<l>`, e.g. `q0=3` in the above example specifies a different `qc` for `l=0` (the s-channel).

Most additional arguments are put inside round brackets. For example, to change the pseudisation scheme to Troullier-Martin you simply put `(tm)`; if you already have a round-bracket section, just put a comma and append what you want, e.g. `(qc=6.2,tm)`.

By default Castep includes the data for the non-linear core corrections (NLCC). You can turn this off by adding `NoNLCC` to the pseudopotential definition string inside the round-bracket section.

You can specify a self-consistent pseudopotential, i.e. one that will be updated after the ground-state calculation to take account of the actual valence charge distribution, by adding `scpsp` to the pseudopotential definition string inside round-brackets.

A pair of square brackets `[]` tells Castep to generate full testing data. For each pseudopotential Castep generates, e.g. `C_OTF.usp`, it will write out several additional files that can be viewed with `xmgrace`:

- `C_OTF.pwave` contains the all-electron and pseudo partial waves
- `C_OTF.beta` contains the pseudopotential beta-projectors
- `C_OTF.econv` contains the convergence of the atomic calculation with cut-off energy

If you want to test the pseudopotential in a different atomic configuration, simply put the configuration inside the square brackets, e.g. `[2s1,2p3]`

Finally, the comment character is a hash `#` – anything on the line after this is ignored.

2.1.5 Symmetry

You can ask Castep to find the symmetry of the simulation cell by adding the keyword `symmetry_generate` to the cell file.

2.2 The Param file

The param file contains the parameters Castep uses for the calculation. The param file is a case-insensitive freeform text file, but unlike the cell file it has no blocks. Each line has the form

```
<parameter> : <value>
```

where `<value>` may be a string, real number, integer, or logical depending on the parameter in question. Logicals accept `'true'`, `'false'`, `'t'` or `'f'`. You may use `'='` instead of a colon if you prefer, and the comment character is `'!'`.

The single most important parameter is `task`, which tells Castep what you want it to do. The most important tasks for EELS-related work are:

- **task** : `SINGLEPOINT` calculates the groundstate energy and charge density for the current cell
- **task** : `GEOMETRYOPTIMISATION` calculates the groundstate energy and charge density for the current cell, and uses the computed forces and stresses to find the groundstate ionic configuration
- **task** : `ELNES` computes the ELNES for the current cell

The fundamental parameters you will need to change are:

- **xc_functional** takes a string, which specifies the exchange-correlation functional, typically LDA or PBE. If you are not generating pseudopotentials on-the-fly you will have to remember to use appropriate pseudopotentials.
- **cut_off_energy** takes a real number, which specifies the plane-wave kinetic-energy cut-off. You may specify an energy unit after the value (default is eV).
- **nextra_bands** takes an integer, which specifies the number of empty bands to include in the ground-state calculation
- **bs_nextra_bands** takes an integer, which specifies the number of empty bands to include in a band-structure/DOS/EELS calculation
- **grid_scale** is a real number which specifies the multiple of the wavefunction grid used for the soft charge density. Ideally this should be 2, but in practice 1.75 or even 1.5 may be sufficient and offers a slight performance improvement; the default is 1.75 for ultrasofts.
- **fine_grid_scale** takes a real number which specifies the multiple of the wavefunction grid used for the hard, augmented charge density. This defaults to **grid_scale**, but may need to be much larger with hard augmentation charges and/or GGAs.

You can shift Castep's optimisation bias to increase speed, or decrease memory usage, via two parameters:

- **opt_strategy_bias** takes an integer from -3 (minimise RAM) to +3 (maximise speed); the default is 0.
- **page_wvfns** takes an integer. If the integer is negative, wavefunctions are usually paged to disk so that only data for a single k-point and spin is held in RAM at any one time. '0' means wavefunctions are *not* usually paged, and a positive integer means wavefunctions should be paged if their RAM requirement exceeds that value in megabytes.

Other commonly used parameters are:

- `spin_polarised` is a logical which tells Castep whether to treat the system as spin-polarised or not. The default is non-spin-polarised unless there are an odd number of electrons in the simulation cell.
- `smearing_width` is a real number specifying the width of the Fermi-level broadening. You may specify an energy unit after the value (default is eV).
- `continuation` takes a string which contains the path and filename of a Castep `check` file, for either restarting a calculation or performing a new type of calculation on a previous result.

As of writing, population analysis does not support core-excited pseudopotentials, so if you are using those you should turn this calculation off using:

- `popn_calculate` is a logical which tells Castep whether or not to perform a Mulliken population analysis of the charge density.

2.3 Getting Help

There are a number of ways to get help in running Castep. If you would like help on a particular set of keywords or functionality, the built-in help is the best place to start, e.g.:

```
[pjh503@bhuna EELS] ./castep -help kpoints_mp_grid
Help information on CELL keywords:
```

```
KPOINTS_MP_GRID    *! SCF Monkhorst-Pack grid !*
```

Type: Integer Vector

Level: Basic

The k-points defined by a Monkhorst-Pack grid. Symmetry (if specified) will be used to generate the SCF k-point list and weights. Default value: Generated from KPOINTS_MP_SPACING.

Help information on PARAMETERS keywords:

None found

You can perform a search on specified text, e.g.

```
[pjh503@bhuna EELS] ./castep -help search band
Help information on CELL keywords:
```

```
BS_KPOINT_PATH *! Band structure path
BS_KPOINT_LIST *! Band structure k-point list
BS_KPOINT_MP_GRID *! Band structure Monkhorst-Pack grid
BS_KPOINT_MP_SPACING *! Band structure Monkurst-Pack grid density
```


Help information on PARAMETERS keywords:

```
WRITE_ORBITALS *! unformatted orbitals file on/off
NLXC_PPD_INTEGRAL *! perform parallelepiped integration
NEXTRA_BANDS *! number of extra bands
PERC_EXTRA_BANDS *! percentage of extra bands
NBANDS *! number of bands
BS_MAX_ITER *! maximum iterations in band structure calculation
BS_MAX_CG_STEPS *! maximum CG steps in band structure calculation
BS_NEXTRA_BANDS *! number of extra bandstructure bands
BS_PERC_EXTRA_BANDS *! percentage of extra bandstructure bands
BS_NBANDS *! number of bands/k-point in band structure calculation
BS_EIGENVALUE_TOL *! band convergence tolerance
BS_XC_FUNCTIONAL *! BS exchange-correlation functional
BS_RE_EST_K_SCRN *! re-estimate k-screening length before BS run
OPTICS_NEXTRA_BANDS *! number of extra optics bands
OPTICS_PERC_EXTRA_BANDS *! percentage of extra bandstructure bands
OPTICS_NBANDS *! number of bands in optics calculation
EXCITED_STATE_SCISSORS *! "scissors" operator band-gap correction
```

which will list all of the Castep keywords that match your string, or are considered relevant.

3 ELNES

In addition to the usual Castep inputs, an ELNES calculation requires an extra input file called `ELNES.param`. This is a temporary state of affairs, and it is intended that the ELNES parameters will be included in the Castep param file in the future.

Although the suffix for the ELNES param file is the same as a Castep param file, it is *not* a freeform file and the parameters must be set out in the order below:

```
-25.0  ! emin    !
65.0   ! emax    !
0.05   ! res     !
0.2    ! gamma   !
0.4    ! gauss   !
.true. ! ifermi  !
.false. ! ioutdir !
.false. ! ioutau  ! output in AU?
```

'!' is the comment character; anything after this on the line is ignored.

The ELNES will be plotted from energy `emin` to `emax` in steps of `res`. If `ifermi` is set to `.true.` then these energies are relative to the Fermi level. The energy units are eV.

The broadening of the spectrum may be Lorentzian (width specified by `gamma>0.0`) or Gaussian (width specified by `gauss>0.0`). If both are nonzero, Lorentzian broadening is applied.

`ioutdir` specifies whether the output should be directionally-resolved or not. If `ioutdir` is `.true.` then three values are written out at each energy; otherwise a spatial average (one-third of the sum) is written.

If `ioutau` is `true` then the output will be written in atomic units, otherwise energies will be given in the same energy units as Castep (default is eV; may be overridden using parameter `energy_unit`).

4 Tutorial - Diamond

Diamond is a useful system to study for ELNES because it is relatively simple, and there are good data available for the EELS, both experimental and *ab initio*.

4.1 2-atom primitive cell

First we need a cell and param file, remembering that we need to generate the pseudopotential on-the-fly. A simple cell file might be:

```
%block LATTICE_CART ! specify the lattice vectors in Angstroms
ang
0.00 1.77 1.77
1.77 0.00 1.77
1.77 1.77 0.00
%endblock LATTICE_CART

%block POSITIONS_FRAC
  C 0.0000000000 0.0000000000 0.0000000000
  C 0.2500000000 0.2500000000 0.2500000000
%endblock POSITIONS_FRAC

symmetry_generate

%BLOCK SPECIES_POT
C 2|1.4|1.4|1.3|6|10|12|20:21(qc=6)
%ENDBLOCK SPECIES_POT

kpoints_mp_grid 10 10 10

bs_kpoints_mp_grid 14 14 14
```

Let us put this in a file called `diamond.cell`. Before we perform an ELNES calculation it is probably wise to relax the simulation cell, so for our first param file we shall choose:

```
task : geometryoptimisation
```

```
cut_off_energy : 400 ev  
xc_functional : lda  
grid_scale : 2.0  
nextra_bands : 8  
bs_nextra_bands : 60
```

```
opt_strategy_bias : +3
```

```
popn_calculate : false
```

Let us put this in a file called `diamond.param`. By default Castep will relax both ionic and cell degrees of freedom, which is what we want. Note that `bs_nextra_bands` is irrelevant here, as we're not performing any band-structure calculations.

To run this calculation simply use

```
[pjh503@bhuna EELS] ./castep diamond &
```

You might like to monitor the main Castep output file `diamond.castep` to check on the calculation's progress.

Once the geometry optimisation has completed successfully, we need to perform an ELNES calculation on the final system. We could copy all the ionic configuration data back by hand into a new cell file, but it is much easier just to tell Castep to continue from where it left off. The following param file is sufficient:

```
task : ELNES  
continuation
```

```
cut_off_energy : 400 ev  
xc_functional : lda  
grid_scale : 2.0  
nextra_bands : 8  
bs_nextra_bands : 60
```

```
opt_strategy_bias : +3
```

```
popn_calculate : false
```

Note that we've used the parameter `continuation` without a value – by default Castep will use the seedname check file, so in our case it will look for `diamond.check`. This will update the simulation cell from the check file so you don't have to amend `diamond.cell`.

In general you will also need an `ELNES.param` file, but if you omit it Castep will choose fairly sensible defaults.

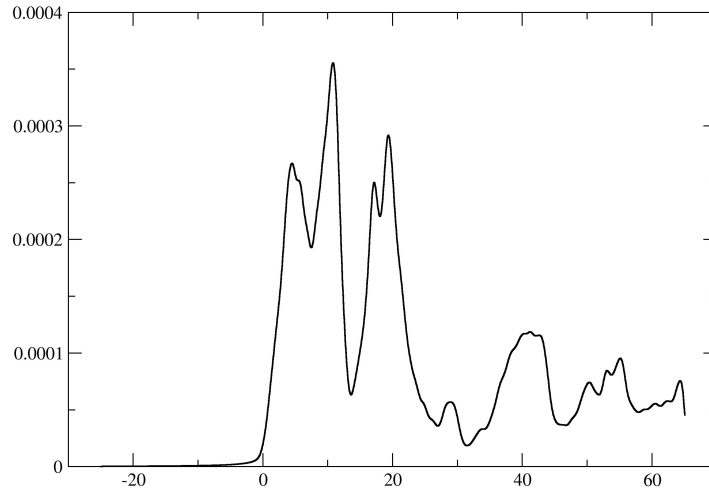


Figure 1: xmgrace output showing the broadened ELNES for a two-atom diamond cell

Rerun castep, and when it has finished you will see `diamond.elnes` appear. You can plot this directly using xmgrace:

```
[pjh503@bhuna EELS] xmgrace diamond.elnes &
```

Since Castep generates a spectrum from each atom, there are in fact two spectra here. The atoms ought to be equivalent though, so in a well-converged calculation the lines should be on top of each other, as in fact they are in this case (see figure 1).

4.2 8-atom FCC cell

Now we will generate a slightly larger, 8-atom diamond cell, using:

```
%block LATTICE_CART ! In Angstroms
ang
3.50000 0.00000 0.00000
0.00000 3.50000 0.00000
0.00000 0.00000 3.50000
%endblock LATTICE_CART
```

```

%block POSITIONS_FRAC
  C 0.000000000 0.000000000 0.000000000
  C 0.000000000 0.500000000 0.500000000
  C 0.500000000 0.500000000 0.000000000
  C 0.500000000 0.000000000 0.500000000
  C 0.750000000 0.250000000 0.750000000
  C 0.250000000 0.250000000 0.250000000
  C 0.250000000 0.750000000 0.750000000
  C 0.750000000 0.750000000 0.250000000
%endblock POSITIONS_FRAC

```

Again, we relax the geometry. You will probably find Castep issues a warning during the search for the ground-state charge density about not having any empty bands. By default Castep treats all simulation systems as if they were metallic, and adds a thermal-like Fermi-level smearing. For this cell the default smearing width (0.2 eV) is large enough that even the top bands are occupied, albeit only slightly, and this can cause problems in the ground-state search.

To avoid these problems we could treat the system as an insulator (use parameter `fix_occupancy : true`, reduce the smearing width or increase the number of bands. Let us increase the number of bands, using:

```
nextra_bands : 16
```

Once the geometry optimisation has completed successfully, we need to run an ELNES calculation as before. Since we have 8 atoms now instead of 2, we should increase the number of extra bands included in the ELNES calculation to achieve the same quality:

```
bs_nextra_bands : 240
```

This will give us an ELNES file as before (see figure 2).

4.3 Core-Excited Calculations

So far we have not included the effect of the core-hole on the EELS result. To do this we need to generate a core-excited pseudopotential for one of the atoms. However we don't want to optimise the geometry for the core-excited case, since we expect the ionic relaxation time to be much longer than the electronic excitation and relaxation time. For this reason we update the ionic and cell coordinates in our cell file so that they reflect the relaxed cell for the non-excited atoms, and choose one atom to be replaced by a core-excited pseudopotential.

```

%block LATTICE_CART ! In Angstroms
ang
3.5284451 0.00000 0.00000
0.00000 3.5284451 0.00000

```

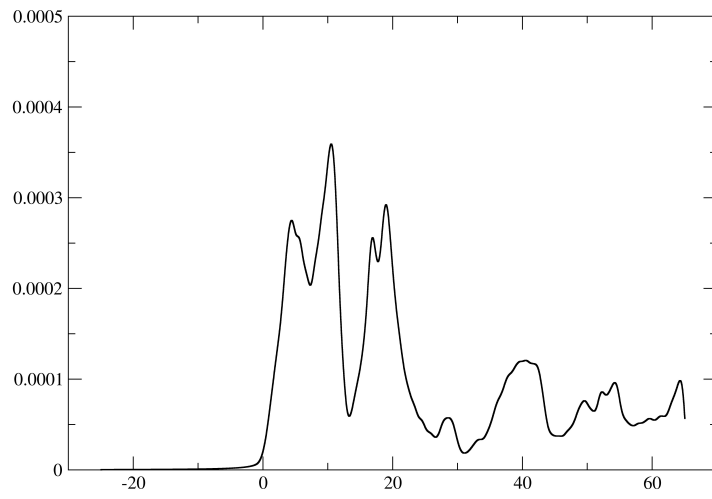


Figure 2: xmgrace output showing the broadened ELNES for an eight-atom diamond FCC cell

```

0.00000 0.00000 3.5284451
%endblock LATTICE_CART

%BLOCK SPECIES_POT
C      2|1.4|1.4|1.3|6|10|12|20:21(qc=6)
C:exc  2|1.4|1.4|1.3|6|10|12|20:21{1s1,2s2,2p3}(qc=6) [] ! add core hole
%ENDBLOCK SPECIES_POT

%block POSITIONS_FRAC
  C:exc  0.0000000000    0.0000000000    0.0000000000
  C      0.0000000000    0.5000000000    0.5000000000
  C      0.5000000000    0.5000000000    0.0000000000
  C      0.5000000000    0.0000000000    0.5000000000
  C      0.7431690000    0.2568310000    0.7431690000
  C      0.2568310000    0.2568310000    0.2568310000
  C      0.2568310000    0.7431690000    0.7431690000
  C      0.7431690000    0.7431690000    0.2568310000
%endblock POSITIONS_FRAC

```

This time we don't want to perform a geometry optimisation, so we jump straight to the ELNES (which is not a continuation this time). Castep will need to perform a ground-state calculation first, but it will do this automatically and then proceed to the EELS stage.

The resulting ELNES file contains spectra due to excitations from each of the atoms in turn; we are only interested in the spectrum from the core-excited atom, so you will need to either edit the ELNES file, or split the data set within `xmgrace`.

The final core-excited spectrum is shown in figure 3, where we have overlaid it on the previous non-excited spectrum for ease of comparison. We can see that the effect of the core-hole is to suppress certain transitions and enhance others.

5 Further Reading

For more information on Castep generally, see <http://www.castep.org>.

For information on (and an example of) the EELS work in Castep see "Cubic boron nitride: Experimental and theoretical energy-loss near-edge structure", *Physical Review B*, volume 64, 115107. More details of the EELS methodology in Castep may be found in Chris Pickard's PhD thesis, which is available online at <http://www.tcm.phy.cam.ac.uk/~cjp20/old/thesis.ps.gz>

A Units

You can specify the units Castep will write the output in, as well as those that the input data is in. For output units you simply add an extra line in the

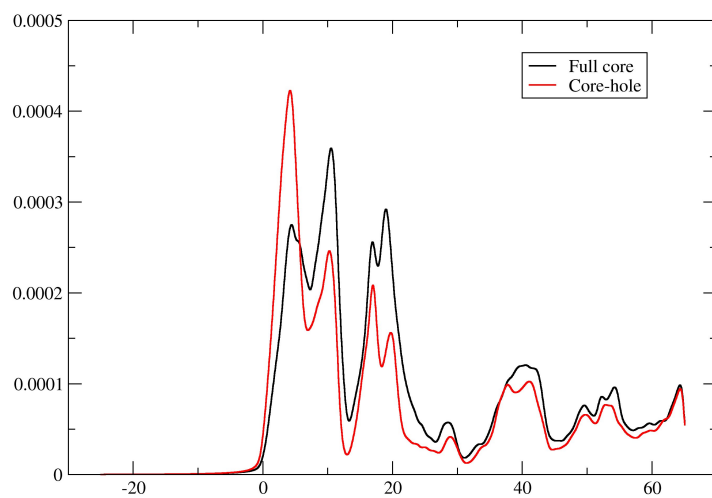


Figure 3: xmgrace output showing the broadened ELNES for an eight-atom diamond FCC cell, both with and without the core-excited pseudopotential

parameter file specifying which quantity you want to change, and which unit you want to use for it, e.g.:

```
energy_unit : eV
```

For the input data the unit is usually specified on the first line of block data, e.g.

```
%block lattice_abc  
ang  
<a> <b> <c>  
<alpha> <beta> <gamma>  
%endblock lattice_abc
```

or after the value itself in single-line data, e.g.

```
cut_off_energy : 12.8 Ry
```

A.1 Length units

The output length unit is controlled by the parameter `length_unit`. Possible values are:

- bohr
- ang (Angstroms; this is the default)
- nm (nanometres)
- cm (centimetres)
- m (metres)

Some parameters have units of reciprocal length, set by the parameter `inv_length_unit`. These can be

- 1/bohr
- 1/ang (inverse Angstroms; this is the default)
- 1/nm (inverse nanometres)
- 1/m (metres)

A.2 Energy units

The output energy unit is controlled by the parameter `energy_unit`. Possible values are:

- Ha or `Hartree` (Hartrees)
- mHa or `mHartree` (milli-Hartrees)
- eV (electron-Volts; this is the default)
- meV (millielectron-Volts)
- Ry (Rydbergs)
- mRy (milli-Rydbergs)
- kJ/mol (kilojoules per mole)
- kcal/mol (kilocalories per mole)
- `erg`
- Hz (Hertz)
- MHz (mega-Hertz)
- GHz (giga-Hertz)
- THz (tera-Hertz)
- `cm-1` (wavenumber)
- K (Kelvin)

Although frequencies have the same dimensions as energies, it is often convenient to use different units. The parameter `frequency_unit` controls the output units used for frequencies. It may take any of the energy values, but the default is `cm-1` (i.e. wavenumber).