

European Theoretical Spectroscopy Facility  
Specification of file formats  
for ETSF Specification version 3.3  
Second revision for this version  
(SpecFF\_ETSF3.3)  
by the Software Integration team and  
collaborators

X. Gonze, C.-O. Almbladh, A. Cucca, D. Caliste, M. Marques,  
C. Freysoldt, V. Olevano, Y. Pouillon, F. Sottile, M. Verstraete

28 September 2008

# Contents

<b>1</b>	<b>General considerations concerning the present file format specifications</b>	<b>6</b>
<b>2</b>	<b>General specifications for ETSF NetCDF files</b>	<b>8</b>
2.1	Global attributes of ETSF NetCDF files . . . . .	8
2.1.1	Detailed description (tables 1 and 2) . . . . .	8
2.2	Generic attributes of variables in ETSF NetCDF files . . . . .	8
2.2.1	Detailed description (table 3) . . . . .	8
2.3	Flag-like attributes . . . . .	9
2.4	Dimensions . . . . .	9
2.4.1	Detailed description (table 4) . . . . .	10
2.4.2	Detailed description (Table 5) . . . . .	11
2.5	Optional variables . . . . .	13
2.5.1	Detailed description (tables 8 to 10) . . . . .	14
2.6	Naming conventions . . . . .	16
<b>3</b>	<b>Specification for files containing crystallographic data</b>	<b>17</b>
3.1	Specification . . . . .	17
3.1.1	Detailed description (table 11) . . . . .	18
3.2	Other file formats for crystallographic data . . . . .	19
<b>4</b>	<b>Specification for files containing a density or a potential</b>	<b>20</b>
4.1	Specification . . . . .	20
4.2	Comments . . . . .	21
4.3	Discussion and future . . . . .	21
<b>5</b>	<b>Specification for files containing the wavefunctions.</b>	<b>23</b>
5.1	Specification . . . . .	23
5.1.1	Detailed description (Table 14) . . . . .	25
5.1.2	Detailed description (table 15) . . . . .	25
5.1.3	Detailed description (tables 16, 17 and 18) . . . . .	26
5.1.4	Detailed description (table 19) . . . . .	28
5.1.5	Detailed description (table 21) . . . . .	29
5.2	Comments . . . . .	29
5.3	Discussion and future . . . . .	32
<b>6</b>	<b>Dielectric function</b>	<b>34</b>
6.1	Specification . . . . .	34
6.1.1	Detailed description (table 22) . . . . .	35
6.1.2	Detailed description (table 23) . . . . .	36
6.1.3	Detailed description (table 24) . . . . .	37
6.1.4	Detailed description (table 25) . . . . .	38
6.1.5	Detailed description (table 26) . . . . .	39
6.1.6	Detailed description (table 27) . . . . .	40

6.1.7	Detailed description (table 28)	41
6.1.8	Detailed description (table 29)	42
6.2	Comments	43
<b>7</b>	<b>Pseudopotential / PAW set up files</b>	<b>44</b>
<b>8</b>	<b>Files with other contents relevant to ETSF</b>	<b>45</b>
<b>9</b>	<b>Appendix A: Some information on the NetCDF size limitation</b>	<b>46</b>
<b>10</b>	<b>Appendix B: List of changes of this version with respect to v2.2 of Jun 4, 2007</b>	<b>47</b>
<b>11</b>	<b>Appendix C: List of things under debate</b>	<b>48</b>
<b>12</b>	<b>Appendix D. List of ETSF NetCDF agreed names (variable, attributes, dimensions, following alphabetical ordering)</b>	<b>49</b>

### **Abstract**

In order to allow softwares to interact and exchange data, file format specifications are mandatory. Widely agreed file format specifications are still lacking in the field of first-principles calculations of material properties.

One of the (numerous) objectives of the European Theoretical Spectroscopy Facility is precisely to specify file formats, for the contents that are relevant to the scientific activity of its constituting nodes. Started in 2004, the present document describes the status of the corresponding work. It includes an inventory of the different contents, discussions of existing formats when relevant, as well as detailed (new) NetCDF specifications, for selected contents (crystallographic/density/wavefunctions). It is hoped that the new specifications will be implemented in many different softwares, or (at least) will be the basis of even better file format specifications.

## Introduction

This document has the goal of informing the members of ETSF, as well as the electronic structure community at large, of the agreed ETSF specifications, in view of further discussions and implementations.

It is to be understood as the present status of file specifications, to be updated on a regular basis. It will be referred to as SpecFF\_ETSF3. It is expected that the ETSF nodes will rely on this document to implement the agreed file formats. Still, there are other file formats on which the specification work has to be done. In particular, it is planned that each year, a workshop should review the existing implementations, and discuss further the specifications. Even the few detailed file format specifications present in this document are subject to revision and improvements. The first version of this document, named SpecFFNQ1 (NQ for Nanoquanta, the precursor of the ETSF), was frozen around June 2006. The current version of the file format, and associated information, can be found at <http://www.etsf.eu/fileformats>.

The document is organized in sections :

- Section 1 presents general considerations concerning the present file format specifications.
- Section 2 presents general specifications concerning ETSF NetCDF file formats.
- Section 3 deals with files containing crystallographic data, and present a rather detailed NetCDF specification, ready for exchange of data among the ETSF nodes. It also briefly presents other existing standardization of files containing crystalline structure and atomic geometries.
- Section 4 deals with files containing density/potential, with the same level of detail.
- Section 5 deals with files containing wavefunctions, with the same level of detail.
- Section 7 deals with pseudopotentials / PAW set up files. It presents the existing specifications, and summarizes the debates and conclusions reached during the Louvain-la- Neuve mini-workshop (see below).
- Section 6 deals with files containing dielectric functions and related data.
- Section 8 is an overview of the other contents relevant for ETSF, and the status of file format specification.

# 1 General considerations concerning the present file format specifications

One has to consider separately the *set of data* to be included in each of different types of files, from their *representation*. Concerning the latter, one encounters simple text files, binary files, XML-structured files, NetCDF files, etc ... It was already decided previously (Nanoquanta meeting Maratea Sept. 2004) to evolve towards formats that deal appropriately with the self- description issue, i.e. XML and NetCDF. The inherent flexibility of these representations will also allow to evolve specific versions of each type of files progressively, and refine earlier working proposals. The same direction has been adopted by several groups of code developers that we know of.

Information on NetCDF and XML can be obtained from the official Web sites:

<http://www.unidata.ucar.edu/software/netcdf/> and

<http://www.w3.org/XML/>

There are numerous other presentations of these formats on the Web, or in books. The elaboration of file formats based on NetCDF has advanced a lot during the Louvain-la- Neuve mini-workshop. There has been also some remarks about XML.

## *Concerning XML*

(A) The XML format is most adapted for the structured representation of relatively small quantity of data, as it is not compressed.

(B) It is a very flexible format, but hard to read in Fortran (no problem in C, C++ or Python). Recently, Alberto Garcia has set up a XMLF90 library of routines to read XML from Fortran.

<http://lcdx00.wm.lc.ehu.es/~wdpgaara/xml/index.html>

Other efforts exists in this direction <http://nn-online.org/code/xml/>

## *Concerning NetCDF*

(A) Several groups of developers inside ETSF have already a good experience of using it, for the representation of binary data (large files).

(B) Although there is no clear advantage of NetCDF compared to HDF (another possibility for large binary files), this experience inside the ETSF network is the main reason for preferring it. By the way, NetCDF and HDF are willing to merge (this might take a few years, though).

(C) File size limitations of NetCDF exist, see appendix D, but should be overcome in the future.

Thanks to the flexibility of NetCDF, the content of a NetCDF file format suitable for use for ETSF softwares might be of four different types :

(1) The actual numerical data (that defines a file for wavefunctions, or a density file, etc ...), whose NetCDF description would have been agreed.

(2) The auxiliary data that are mandatory to make proper usage of the actual numerical data. The NetCDF description of these auxiliary data should also be agreed.

(3) The auxiliary data that are not mandatory, but whose NetCDF description

has been agreed, in a larger context.

(4) Other data, typically code-dependent, whose existence might help the use of the file for a specific code. The name of these variables should be different from the names chosen for agreed variables (1)-(3). Such other data might even be redundant with (1)-(3).

Such content is compatible with a file format being complete for use by many codes, though adapted for the specific usage by one code. The ETSF file descriptions to be provided later (sections 2 to 5) are based on this generic classification of data that can be integrated in such a NetCDF file.

In order to address the 2 GB limit (see Appendix F), as well as the use of NetCDF files for parallel calculations, one file can actually be split into several partial files. Selected variables should describe the differing content of each of them. As an example, in section 4, a file containing a set of wavefunctions can be split in different files containing selected bands and/or k-points, however being exactly similar in every other respect.

Some technical details concerning the use of NetCDF files will apply to all formats specified in the ETSF framework :

1. concerning the variable names, long names should be chosen, as close as possible to natural language (so inherently self-descriptive).
2. all variable names are lower case, except “Conventions” - a name agreed by the NetCDF community
3. underscores are used to replace blanks separating words
4. in the tables, the slow indices are left-most, and the fast indices are right-most, so that the *order of indices has to be reversed* in FORTRAN

## 2 General specifications for ETSF NetCDF files

### 2.1 Global attributes of ETSF NetCDF files

Global attributes are used for a general description of the file, mainly the file format convention. Important data is not contained in attributes, but rather in variables.

The length of character attributes is the maximum length this attribute may take. This is relevant for reading, where sufficient space must be provided. In writing, the defined length may be reduced to the real length of the attribute.

Table 1 gathers specifications for required attributes in any ETSF NetCDF files. Table 2 presents optional attributes for ETSF NetCDF files.

Attributes	Type (length)	Notes
file_format	char (80)	“ETSF”
file_format_version	real	1.1 or 1.2 or 2.0 ...
Conventions	char (80)	“ <a href="http://www.etsf.eu/fileformats">http://www.etsf.eu/fileformats</a> ”

Table 1: Mandatory global attributes for ETSF NetCDF files

Attributes	Type (length)	Notes
history	char (1024)	
title	char (80)	

Table 2: Optional global attributes for ETSF NetCDF files.

#### 2.1.1 Detailed description (tables 1 and 2)

**file\_format** Name of the file format for ETSF wavefunctions.

**file\_format\_version** Real version number for file format (only one period, e.g. 1.2 ). Conventions NetCDF recommended attribute specifying where the conventions for the file can be found on the Internet.

**history** NetCDF recommended attribute : each code modifying/writing this file is encouraged to add a line about itself in the history attribute. char(1024) allows for 12 additions of at most 80 characters.

**title** Short description of the content (system) of the file.

### 2.2 Generic attributes of variables in ETSF NetCDF files

A few attributes might apply to a large number of variables. They are gathered in Table 3.

#### 2.2.1 Detailed description (table 3)

**units** It is one of the NetCDF recommended attributes, but it only applies to a few variables in our case, since most are dimensionless. For dimensional

Attributes	Type (length)	Notes
units	char (80)	required for variables that carry units
scale_to_atomic_units	double	required for units other than “atomic units”

Table 3: Generic attributes that might be mandatory for selected variables in ETSF NetCDF files.

variables, it is required. The use of atomic units (corresponding to the string “atomic units”) is advised throughout for portability. If other units are used, the definition of an appropriate scaling factor to atomic units is mandatory. Actually, the definition of the name “units” in the ETSF files is only informative : the “scale\_to\_atomic\_units” information should be the only one used to read the file by machines.

**scale\_to\_atomic\_units** If “units” is something other than the character string “atomic units” (based on Hartree for energies, Bohr for lengths) we request the definition of an appropriate scaling factor. The appropriate value in atomic units is obtained by multiplying the number found in the variable by the scaling factor. Examples:

units=“eV” → scale\_to\_atomic\_units = 0.036749326

units=“angstrom” → scale\_to\_atomic\_units = 1.8897261

units=“parsec” → scale\_to\_atomic\_units = 5.8310856e+26

This can be used to deal with unknown units. Note that the recommended values for the fundamental constants can be found at <http://physics.nist.gov/cuu/Constants/index.html>

## 2.3 Flag-like attributes

“Flag-like” attributes can take the values “yes” and “no”. When such attributes are written, they should be written in full length and small letters. When they are read, only the first character needs to be checked (i.e. “y” or “n” – this simplifies life a lot).

## 2.4 Dimensions

Dimensions are used for one- or multidimensional variables. It is very important to remember that the NetCDF interface adapts the dimension ordering to the programming language used. The notation here is C-like, i.e. the last index varies fastest. In Fortran, the order is reversed. When implementing new reading interfaces, the dimension names can be used to check the dimension ordering. The dimension names help also to identify the meaning of certain dimensions in cases where the number alone is not sufficient.

The list of variables that specify dimensions in ETSF NetCDF files is given in Table 4 and 5. Table 4 list the dimensions that are not supposed to lead to a splitting, while table 5 list the dimensions that might be used to define a splitting (e.g. in case of parallelism).

Dimensions	Type (index order as in C)	Notes
character_string_length	integer	Always ==80
real_or_complex_coefficients	integer	Either ==1 or 2
real_or_complex_density	integer	Either ==1 or 2
real_or_complex_gw_corrections	integer	Either ==1 or 2
real_or_complex_potential	integer	Either ==1 or 2
real_or_complex_wavefunctions	integer	Either ==1 or 2
number_of_cartesian_directions	integer	Always ==3
number_of_reduced_dimensions	integer	Always ==3
number_of_vectors	integer	Always ==3
number_of_symmetry_operations	integer	
number_of_atoms	integer	
number_of_atom_species	integer	
symbol_length	integer	Always ==2

Table 4: Variables that specify dimensions in ETSF NetCDF files (no splitting case).

#### 2.4.1 Detailed description (table 4)

**character\_string\_length** The maximum length of string variables (attributes may be longer).

**real\_or\_complex\_coefficients** To specify whether the variable coefficients\_of\_wavefunctions (table 16) is real or complex

**real\_or\_complex\_density** To specify whether the variable density (table 12) is real or complex

**real\_or\_complex\_gw\_corrections** To specify whether the variable gw\_corrections (table 21) is real or complex

**real\_or\_complex\_potential** To specify whether the variables exchange\_potential, correlation\_potential, and exchange\_correlation\_potential (table 13) are real or complex

**real\_or\_complex\_wavefunctions** To specify whether the variable real\_space\_wavefunctions (table 19) is real or complex .

**number\_of\_cartesian\_directions** Used for absolute coordinates.

**number\_of\_reduced\_dimensions** Used for reduced (also called relative) coordinates in reciprocal or real space.

**number\_of\_vectors** Used to distinguish the vectors when defining their relative / reduced coordinates.

**number\_of\_symmetry\_operations** The number of symmetry operations.

**number\_of\_atoms** The number of atoms in the unit cell.

**number\_of\_atom\_species** The number of different atom species in the unit cell.

**symbol\_length** Maximum number of characters for the chemical symbols

Dimensions	Type (index order as in C)	Notes
max_number_of_states	integer	
number_of_kpoints	integer	
number_of_spins	integer	Either ==1 or 2
number_of_spinor_components	integer	Either ==1 or 2
number_of_components	integer	Either ==1, 2 or 4
max_number_of_coefficients	integer	
number_of_grid_points_vector1	integer	
number_of_grid_points_vector2	integer	
number_of_grid_points_vector3	integer	
max_number_of_basis_grid_points	integer	For wavelets. range in 1 to number_of_grid_points1_vector1* number_of_grid_points1_vector2* number_of_grid_points1_vector3
number_of_localisation_regions	integer	Always 1.

Table 5: Variables that specify dimensions in ETSF NetCDF files (possible splitting case). For the auxiliary variables needed in case of splitting, see Table 7

#### 2.4.2 Detailed description (Table 5)

**max\_number\_of\_states** The maximum number of states

**number\_of\_kpoints** The number of kpoints

**number\_of\_spins** Used to distinguish collinear spin-up and spin-down components :

1 for non-spin-polarized or spinor wavefunctions

2 for collinear spin (spin-up and spin-down)

**number\_of\_spinor\_components** For non-spinor wavefunctions, this dimension must be present and equal to 1. For spinor wavefunctions this dimension must equal to 2.

**number\_of\_components** Used for the spin components of spin-density matrices :

1 for non-spin-polarized

2 for collinear spin (spin-up and spin-down)

4 for non-collinear spin (average density, then magnetization vector in cartesian coordinates x,y and z)

**max\_number\_of\_coefficients** The (maximum) number of coefficients for the basis functions at each k-point, except in the case of real space grids (see next lines)

**number\_of\_grid\_points\_vector1** The number of grid points along direction 1 in the unit cell in real space, for dimensioning the wavefunction coefficients (an alternative to max\_number\_of\_coefficients)

**number\_of\_grid\_points\_vector2** Same as number\_of\_grid\_points\_vector1, for the second direction.

**number\_of\_grid\_points\_vector3** Same as `number_of_grid_points_vector1`, for the third direction.

**max\_number\_of\_basis\_grid\_points** For wavelets. The number of relevant points from the regular mesh in real space used to store coefficients. This value is the maximum of number of basis set grid points over all localization region(s). Currently, `number_of_localization_regions` is always 1, so `max_number_of_basis_grid_points` is simply the number of basis set grid points.

**number\_of\_localization\_regions** For wavelets. This dimension will be used later to define one or several localized basis sets.

To clarify the interplay between `number_of_spins`, `number_of_components`, and `number_of_spinor_components`, note the different following magnetic or non-magnetic cases:

Non-spin-polarized :

`number_of_spins=1` , `number_of_spinor_components=1` , `number_of_components=1`

Collinear spin-polarized :

`number_of_spins=2` , `number_of_spinor_components=1` , `number_of_components=2`

Non-collinear spin-polarized :

`number_of_spins=1` , `number_of_spinor_components=2` , `number_of_components=4`

We now turn to the specification of the (optional) splitting of files in partial files. Such splitting might be done in many different ways. In order to allow for very general, flexible, splittings, but still rely on a simple system, we set up different pairs of variables, one for each possible splitting. These pairs of variables are described in Table 2.6. If a software cannot cope with the file splitting, it should simply check that no file splitting is done, and if the contrary happens, it should stop.

Let us work out an example.

Suppose we split the file according to the `kpoints`. The full set might have 10 `kpoints`, of which 3 `kpoints` (number 1, 2 and 5) might be contained in a first file, 3 other `kpoints` (number 3, 6 and 9) might be contained in a second file, and the 4 remaining `kpoints` (number 4, 7, 8 and 10) might be contained in the third file.

Then, the first file will contain :

`number_of_kpoints = 10` , `my_number_of_kpoints = 3` , `my_kpoints=(1,2,5)`

The second file will contain :

`number_of_kpoints = 10` , `my_number_of_kpoints = 3` , `my_kpoints=(3,6,9)`

The third file will contain :

`number_of_kpoints = 10` , `my_number_of_kpoints = 4` , `my_kpoints=(4,7,8,10)`

If more than one splitting is done, the file will contain the intersection of the split data. As an example, suppose we split the file according to the `kpoints` and the `spins`. The full set of `kpoints` might have 4 `kpoints`, and there would be two `spins`. We perform two splittings, one separating `kpoints` 1 and 2 from `kpoints` 3 and 4, and one separating the `spins`.

The first file might contain :

`number_of_kpoints = 4` , `my_number_of_kpoints = 2` , `my_kpoints=(1,2)`

`number_of_spins = 2` , `my_number_of_spins = 1` , `my_spins=(1)`

The second file might contain :  
number\_of\_kpoints = 4 , my\_number\_of\_kpoints = 2 , my\_kpoints=(3,4)  
number\_of\_spins = 2 , my\_number\_of\_spins = 1 , my\_spins=(1)

The third file might contain :  
number\_of\_kpoints = 4 , my\_number\_of\_kpoints = 2 , my\_kpoints=(1,2)  
number\_of\_spins = 2 , my\_number\_of\_spins = 1 , my\_spins=(2)

The fourth file might contain :  
number\_of\_kpoints = 4 , my\_number\_of\_kpoints = 2 , my\_kpoints=(3,4)  
number\_of\_spins = 2 , my\_number\_of\_spins = 1 , my\_spins=(2)

Different variables might change their sizes when splitting is used. The list of variables whose size might change compared to non-split files will have to be specified.

Dimensions	Type (index order as in C)	Notes
my_max_number_of_states	integer	At least 1, at most number_of_states
my_number_of_kpoints	integer	At least 1, at most number_of_kpoints
my_number_of_spins	integer	At least 1, at most number_of_spins
my_number_of_spinor_components	integer	At least 1, at most number_of_spinor_components
my_number_of_components	integer	At least 1, at most number_of_components
my_number_of_grid_points_vector1	integer	At least 1, at most number_of_grid_points_vector1
my_number_of_grid_points_vector2	integer	At least 1, at most number_of_grid_points_vector2
my_number_of_grid_points_vector3	integer	At least 1, at most number_of_grid_points_vector3
my_max_number_of_coefficients	integer	At least 1, at most max_number_of_coefficients

Table 6: Dimensions of variables to specify the (optional) splitting of one file in different partial files. These dimensions and associated variables (see Table 7) are defined by pair (one integer, and one integer array). Any one of these pairs can be used to split the files, and several of these pairs can be used as well. In case several pairs are used, the content of the file is defined by the intersection of the different integer arrays. The detailed description of these variables is induced from the one of the corresponding Table 5 variables.

## 2.5 Optional variables

In order to avoid the “divergence of the formats in the additional data” (Olevano), we propose names and formats for some information that is likely to be

Variables	Type (index order as in C)	Notes
my_states	integer [my_max_number_of_states]	
my_kpoints	integer [my_number_of_kpoints]	
my_spins	integer [my_number_of_spins]	
my_spinor_components	integer [my_number_of_spinor_components]	
my_components	integer [my_number_of_components]	
my_grid_points_vector1	integer [my_number_of_grid_points_vector1]	
my_grid_points_vector2	integer [my_number_of_grid_points_vector2]	
my_grid_points_vector3	integer [my_number_of_grid_points_vector3]	
my_coefficients	integer [my_max_number_of_coefficients]	

Table 7: Variables to specify the (optional) splitting of one file in different partial files. See the explanation in Table 6. The detailed description of these variables is induced from the one of the corresponding Table 5 variables.

written to the files. This section will grow in future format versions. Please report any variable you miss here, so we can add it to the list. None of these data is mandatory for the file formats to be described later. Some of the proposed variables contain redundant information.

All optional variables must be defined BEFORE the largest size array of the file, otherwise this array will be restricted to 4GB. Examples of such arrays are `coefficients_of_wavefunctions` or `real_space_wavefunctions` (see later).

Tables 8 to 10 present these optional variables, grouped with respect to their physical relevance : atomic information, electronic structure, and reciprocal space.

Variables	Type (index order as in C)	Notes
valence_charges	double [number_of_atom_species]	
pseudopotential_types	char [number_of_atom_species] [character_string_length]	

Table 8: Optional variables : atomic information

### 2.5.1 Detailed description (tables 8 to 10)

**valence\_charges** Ionic charges for each atom species.

**number\_of\_electrons** Number of electrons in the elementary cell.

**fermi\_energy** Fermi energy corresponding to occupation numbers.

Variables	Type (index order as in C)	Notes
number_of_electrons	integer	
exchange_functional	char [character_string_length]	
correlation_functional	char [character_string_length]	
fermi_energy	double	Units attribute required. The attribute "scale_to_atomic_units" might also be mandatory, see section 2.2.
smearing_scheme	char [character_string_length]	
smearing_width	double	Units attribute required. The attribute "scale_to_atomic_units" might also be mandatory, see section 2.2.

Table 9: Optional variables : electronic structure

Variables	Type (index order as in C)	Notes
kinetic_energy_cutoff	double	Units attribute required. The attribute "scale_to_atomic_units" might also be mandatory, see section 2.2.
kpoint_grid_shift	double [number_of_reduced_dimensions]	
kpoint_grid_vectors	double [number_of_vectors] [number_of_reduced_dimensions]	
monkhorst_pack_folding	integer [number_of_vectors]	

Table 10: Optional variables : reciprocal space

**smearing\_scheme** Smearing scheme used for metallic or finite temperature occupation numbers = "gaussian", "fermi-dirac", "cold-smearing", "methfessel-paxton-n" for n=1 ... 10

**smearing\_width** Smearing width used with scheme above.

**kinetic\_energy\_cutoff** Cutoff used to generate the plane-wave basis set.

**kpoint\_grid\_vectors** Basis vectors for kpoint grid if it is homogeneous. Given in the coordinates of reciprocal space primitive vectors.

**kpoint\_grid\_shift** Shift for offset of grid of kpoints. Used with both kpoint\_grid\_vectors and monkhorst\_pack\_folding.

**monkhorst\_pack\_folding** This indicates the "folding" for regular kpoint grids (e.g. Monkhorst-Pack Phys. Rev. B 13, 5188 (1976)). An alternative to kpoint\_grid\_vectors.

**pseudopotential\_types** Type of pseudopotential scheme = "bachelet-hamann-schluter", "troullier-martins", "hamann", "hartwigsen-goedecker-hutter", "goedecker-teter-hutter" ... A standardized list should be found or established.

**exchange\_functional** String describing the functional used for exchange:

names should be taken from the ETSF XC library specifications (at present, under construction).

**correlation\_functional** String describing the functional used for correlation: Lee Yang Parr or Colle-Salvetti etc... names should be taken from the ETSF XC library specifications.

## 2.6 Naming conventions

NetCDF files, that respect the ETSF specifications described in the present document, should be easily recognized. We suggest to append, in their names, the string “-etsf.nc” . The appendix “.nc” is a standard convention for naming NetCDF files, see:

<http://www.unidata.ucar.edu/software/netcdf/docs/faq.html#filename> . Some filesystems are case- insensitive, and this motivates the lower-case choice. Finally, a dash is to be preferred to an underscore to allow the files references by a Web search engine.

## 3 Specification for files containing crystallographic data

### 3.1 Specification

A ETSF NetCDF file for crystallographic data should contain the following set of mandatory information :

1. The three attributes defined in Table 1
2. The following dimensions from Table 4 (dimensions that do not lead to a splitting) :
  - number\_of\_cartesian\_directions
  - number\_of\_vectors
  - Note : suppressed complex\_dimension
  - number\_of\_atoms
  - number\_of\_atom\_species
  - number\_of\_symmetry\_operations
3. The following variables defined in Table 11 :
  - primitive\_vectors
  - reduced\_symmetry\_matrices
  - reduced\_symmetry\_translations
  - space\_group
  - atom\_species
  - reduced\_atom\_positions
4. At least one of the following variables defined in Table 11, to specify the kind of atoms :
  - atomic\_numbers
  - atom\_species\_names
  - chemical\_symbols

The use of “atomic\_numbers” is preferred. If “atomic\_numbers” is not available, “atom\_species\_names” will be preferred over “chemical\_symbols”. In case more than one such variables are present in a file, the same order of preference should be followed by the reading program.

As mentioned in section 1 and 2, such file might contain additional information agreed within ETSF, such as any of the variables specified in section 2. It might even contain enough information to be declared a ETSF NetCDF file “containing the density” or “containing the wavefunctions”, or both. Such file might also contain additional information specific to the software that generated

the file. It is not expected that this other software-specific information might be used by another software.

It is not expected that the above-mentioned information might be distributed among different files (unlike for density/potential/wavefunction files, see later).

Variables	Type (index order as in C)	Notes
primitive_vectors	double [number_of_vectors] [number_of_cartesian_directions]	By default, given in Bohr
reduced_symmetry_matrices	integer [number_of_symmetry_operations] [number_of_reduced_dimensions] [number_of_reduced_dimensions]	The “symmorphic” attribute is needed.
reduced_symmetry_translations	double [number_of_symmetry_operations] [number_of_reduced_dimensions]	The “symmorphic” attribute is needed.
space_group	integer	Between 1 and 232
atom_species	integer [number_of_atoms]	Between 1 and “number_of_atom_species”
reduced_atom_positions	double [number_of_atoms] [number_of_reduced_dimensions]	
atomic_numbers	double [number_of_atom_species]	
atom_species_names	char [number_of_atom_species] [character_string_length]	
chemical_symbols	char [number_of_atom_species] [symbol_length]	
Attributes	Type	
symmorphic	char(80)	flag-type attribute, see section 2.3

Table 11: Variables and attributes to specify the atomic structure and symmetry operations.

### 3.1.1 Detailed description (table 11)

**primitive\_vectors** The primitive vectors, expressed in cartesian coordinates.

Symmetry operations are defined in real space, with reduced coordinates. A symmetry operation in real space sends the input point  $r$  to the output point  $r'$ , with

$$r'_{\alpha}{}^{red} = \sum_{\beta} S_{\alpha\beta}{}^{red} r_{\beta}{}^{red} + t_{\beta}{}^{red} \quad (1)$$

The array **reduced\_symmetry\_matrices** contains the matrices  $S$ , in reduced coordinates, of Table 11, while the vector  $t$ , in reduced coordinates, is contained in the array **reduced\_symmetry\_translations** of the same Table. There might be a confusion between the two dimensions “number\_of\_reduced\_dimensions”

of this variable. In the C ordering, the last one corresponds to the beta index in the above-mentioned formula.

The first symmetry operation must always be unity with translation vector (0,0,0). If all translations are zero, the attribute **symmorphic** for `reduced_symmetry_matrices` should be set to "yes".

**space\_group** Space group number according to international tables of crystallography (from 1 to 232)

**atom\_species** Types of each atom in the unit cell. Note that the first type of atom has number "1", and the last type of atom has number "number\_of\_atom\_species".

**reduced\_atom\_positions** Positions of the different atoms in the unit cell in relative / reduced coordinates.

**atomic\_numbers** Atomic number for each atom species. If it does not refer to an "usual" atom (e.g. fractional charge atoms or similar), a non-integer number or zero may be used, but it is strongly advised then to also specify the `atom_species_names` variable.

**atom\_species\_names** Descriptive name for each atom species = "H" "Ga" plus variants like "Ga-semicore" "C-1s-corehole" "C-sp2" "C1"

**chemical\_symbols** Chemical symbol for each atom species (as in periodic table). If not appropriate (fractional charge atoms or similar), "X" may be used.

**symmorphic** Flag-type attribute (see section 2.3), needed for the variables `reduced_symmetry_matrices` and `reduced_symmetry_translations`.

## 3.2 Other file formats for crystallographic data

They were not discussed in the Louvain-la-Neuve meeting. We nevertheless give a list of Web sites, for further work ...

1. CML (Chemical Markup Language) is a metalanguage, based on XML, that allows to specify crystalline structures.  
Original site : <http://www.xml-cml.org>  
CMLCore : <http://wmm.ch.cam.ac.uk/moin/CmlCore>
2. OpenBabel is a tool to convert files describing chemical species from one format to another.  
[http://openbabel.sourceforge.net/wiki/index.php/Main\\_Page](http://openbabel.sourceforge.net/wiki/index.php/Main_Page)  
Many different I/O format for chemical species are mentioned at:  
<http://openbabel.sourceforge.net/wiki/index.php/Babel>

## 4 Specification for files containing a density or a potential

### 4.1 Specification

A ETSF NetCDF file for a density should contain the following set of mandatory information :

1. The three attributes defined in Table 1
2. The following dimensions from Table 4 (dimensions that do not lead to a splitting) :
  - number\_of\_cartesian\_directions
  - number\_of\_vectors
  - real\_or\_complex\_density and/or real\_or\_complex\_potential
3. The following dimensions from Table 5 (dimensions that might lead to a splitting) :
  - number\_of\_components
  - number\_of\_grid\_points\_vector1
  - number\_of\_grid\_points\_vector2
  - number\_of\_grid\_points\_vector3
4. The primitive vectors of the cell, as defined in Table 11
5. The density or potential, as defined in Table 12 or 13 . This variable must be the last, in order not to be limited to 4 GB.

As mentioned in section 1 and 2, such file might contain additional information agreed within ETSF, such as any of the variables specified in section 2. It might even contain enough information to be declared a ETSF NetCDF file “containing crystallographic data” or “containing the wavefunctions”, or both. Such file might also contain additional information specific to the software that generated the file. It is not expected that this other software-specific information might be used by another software.

The information might distributed among different files, thanks to the use of splitting of data for variables :

- number\_of\_components
- number\_of\_grid\_points\_vector1
- number\_of\_grid\_points\_vector2
- number\_of\_grid\_points\_vector3

In case the splitting related to one of these variables is activated, then the corresponding variables in Table 2.6 must be defined. Accordingly, the dimensions of the variables in Table 12 and 13 will be changed, to accommodate only the segment of data effectively contained in the file.

A ETSF NetCDF exchange, correlation, or exchange-correlation potential file should contain at least one variable among the three presented in Table 13 in replacement of the specification of the density. The type and size of such variables are similar to the one of the density. The other variables required for a density are also required for a potential file. Additional ETSF or software-specific information might be added, as described previously.

In case the splitting related to one of these variables is activated, then the corresponding variables in Table 2.6 must be defined. Accordingly, the dimensions of the variables in Tables 12, and/or 13 might have to be changed, to accommodate only the segment of data effectively contained in the file.

Variables	Type (index order as in C)	Notes
density	double[number_of_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex_density]	This is a pseudo-density. Note in case of PAW, the augmentation contribution is missing. By default, the density is given in atomic units, that is, number of electrons per Bohr <sup>3</sup> The “units” attribute is required. The attribute “scale_to_atomic_units” might also be mandatory, see section 2.2.

Table 12: The specification of the density (the last of the variables)

## 4.2 Comments

1. A density in such a format (represented on a 3D homogeneous grid) is suited for the representation of smooth densities, as obtained naturally from pseudopotential calculations using plane waves.
2. The definition of names for other types of potentials meet some problems, that should be examined in further ETSF meetings. In the absence of a specification of name, the ETSF groups are nevertheless encouraged temporarily to define potentials using the same types and sizes, as well as to choose long names.

## 4.3 Discussion and future

1. This specification should be extended to the PAW as well as LAPW methods. We suggest to supplement the present set of variables with other variables, not fixed at present.

Variables	Type (index order as in C)	Notes
correlation_potential	double[number_of_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex_potential]	Note in case of PAW, the augmentation contribution is missing. Units attribute required. The attribute "scale_to_atomic_units" might also be mandatory, see section 2.2.
exchange_potential	double[number_of_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex_potential]	Note in case of PAW, the augmentation contribution is missing. Units attribute required. The attribute "scale_to_atomic_units" might also be mandatory, see section 2.2.
exchange_correlation_potential	double[number_of_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex_potential]	Note in case of PAW, the augmentation contribution is missing. Units attribute required. The attribute "scale_to_atomic_units" might also be mandatory, see section 2.2.

Table 13: The specification of exchange, correlation, and exchange-correlation potentials.

2. The specification of a density, Table 12, can accommodate the response densities of Density-Functional Perturbation Theory.

## 5 Specification for files containing the wavefunctions.

### 5.1 Specification

A ETSF NetCDF file “containing the wavefunctions” should contain at least the information needed to build the density from this file. Also, since the eigenvalues are intimately linked to eigenfunctions, it is expected that such a file contain eigenvalues. Of course, files might contain less information than the one required, but still follow the naming convention of ETSF. It might also contain more information, of the kind specified in other tables of the present document.

A ETSF NetCDF file “containing the wavefunctions” should contain the following set of mandatory information :

1. The three attributes defined in Table 1
2. The following dimensions from Table 4 (dimensions that do not lead to a splitting) :
  - `character_string_length`
  - `number_of_cartesian_directions`
  - `number_of_vectors`
  - `real_or_complex_coefficients` and/or `real_or_complex_wavefunctions`
  - `number_of_symmetry_operations`
  - `number_of_reduced_dimensions`
3. The following dimensions from Table 5 (dimensions that might lead to a splitting) :
  - `max_number_of_states`
  - `number_of_kpoints`
  - `number_of_spins`
  - `number_of_spinor_components`
4. In case of a real-space wavefunctions, the following dimensions from Table 5 :
  - `number_of_grid_points_vector1`
  - `number_of_grid_points_vector2`
  - `number_of_grid_points_vector3`
  - \* or, in case of a wavefunction given in terms of a basis set, the following dimensions from Table 5:
    - `max_number_of_coefficients`

- \* or in case of a wavefunction given in terms of a Daubechies wavelet basis set, the following dimensions from Table 5 :
  - max\_number\_of\_basis\_grid\_points
  - number\_of\_localization\_regions
5. The primitive vectors of the cell, as defined in Table 11 (variable primitive\_vectors)
  6. The symmetry operations, as defined in Table 11 (given by the variables reduced\_symmetry\_translations and reduced\_symmetry\_matrices)
  7. The information related to each kpoint, as defined in Table 14
  8. The information related to each state (including eigenenergies and occupation numbers), as defined in Table 15
  9. In case of basis set representation, the information related to the basis set, and the variable coefficients\_of\_wavefunctions, as defined in Table 16. For basis\_set equals to “plane\_waves”, the following variable are required from Table 16 :

- reduced\_coordinates\_of\_plane\_waves

When basis\_set equals “daubechies\_wavelets”, the following variables are required from Table 16 :

- coordinates\_of\_basis\_grid\_points
- number\_of\_coefficients\_per\_grid\_point

The description of the Daubechies wavelet basis set is detailed in section 5.2 “comments”, point (3).

10. In case of real-space representation, the variable real\_space\_wavefunctions, see Table 19. In order not to be limited to 4 GB, the variables coefficients\_of\_wavefunctions or real\_space\_wavefunctions must be the last on.

As mentioned in section 1 and 2, such a file might contain additional information agreed on within ETSF, such as any of the variables specified in section 2. It might even contain enough information to be declared a ETSF NetCDF file “containing crystallographic data” or “containing the density”, or both. Such a file might also contain additional information specific to the software that generated the file. It is not expected that this other software-specific information might be used by another software.

The information might be distributed among different files, thanks to the use of splitting of data for variables :

- max\_number\_of\_states
- number\_of\_kpoints

- `number_of_spins`

And, either

- `number_of_grid_points_vector1`
- `number_of_grid_points_vector2`
- `number_of_grid_points_vector3`

or

- `max_number_of_coefficients`

In case the splitting related to one of these variables is activated, then the corresponding variables in Table 2.6 must be defined. Accordingly, the dimensions of the variables in Tables 14 to 19 might have to be changed, to accommodate only the segment of data effectively contained in the file.

Variables	Type (index order as in C)	Notes
<code>reduced_coordinates_of_kpoints</code>	<code>double[number_of_kpoints] [number_of_reduced_dimensions]</code>	See possible changes for split files in Table 20
<code>kpoint_weights</code>	<code>double[number_of_kpoints]</code>	See description of density construction in section 5.2 See also possible changes for split files in Table 20

Table 14: Variables that specify the k points

### 5.1.1 Detailed description (Table 14)

**reduced\_coordinates\_of\_kpoints** k-point in relative / reduced coordinates

**kpoint\_weights** k-point integration weights. The weights must sum to 1. See the description of the density construction, section 5.2.

### 5.1.2 Detailed description (table 15)

**number\_of\_states** Number of states for each kpoint, if varying (the attribute `k_dependent` must be set to yes). Otherwise (the attribute `k_dependent` must be set to no), might not contain any information, the actual number of states being set to `max_number_of_states`.

**eigenvalues** One-particle eigenvalues/eigenenergies. Should be 0 if unknown.

**occupations** Occupation numbers. Full occupation for spin-unpolarized cases (`number_of_spins = 1` AND `number_of_spinor_components = 1`) is 2, otherwise it is 1. See section 5.2.

**k\_dependent** Flag-type attribute (see section 2.3), needed for the variables `number_of_states`, `number_of_coefficients`, and `reduced_coordinates_of_plane_waves`.

Variables	Type (index order as in C)	Notes
number_of_states	integer[number_of_spins] [number_of_kpoints]	the attribute "k_dependent" must be defined
eigenvalues	double[number_of_spins] [number_of_kpoints] [max_number_of_states]	The "units" attribute is required. The attribute "scale_to_atomic_units" might also be mandatory, see section 2.2. See also possible changes for split files in Table 20
occupations	double[number_of_spins] [number_of_kpoints] [max_number_of_states]	See also possible changes for split files in Table 20
Attributes	Type (index order as in C)	
k_dependent	char(80)	attribute of number_of_states, flag-type, see section 2.3

Table 15: Specifications related to each state : occupation numbers and eigenvalues

### 5.1.3 Detailed description (tables 16, 17 and 18)

**basis\_set** Type of basis set used if not in a real-space grid. At present, either "plane\_waves" or "Daubechies\_wavelets".

**number\_of\_coefficients** Number of basis function coefficients for each kpoint, if varying (the attribute k\_dependent must be set to yes). Otherwise (the attribute k\_dependent must be set to no), might not contain any information, the actual number of coefficients being set to max\_number\_of\_coefficients.

**reduced\_coordinates\_of\_plane\_waves** Plane-wave G-vectors in relative / reduced coordinates. If the attribute k\_dependent is set to no, then the dimension [number\_of\_kpoints] must be omitted. On the other hand, if the attribute used\_time\_reversal\_at\_gamma is set to yes (only allowed for the plane wave basis set), then, for the Gamma k point - reduced\_coordinates\_of\_kpoints being equal to (0 0 0) - the time reversal symmetry has been used to nearly halve the number of plane waves, with the coefficients of the wavefunction for a particular reciprocal vector being the complex conjugate of the coefficients of the wavefunction at minus this reciprocal vector. So, apart the origin, the coefficient of only one out of each pair of corresponding plane waves ought to be specified. Note that in the present version of this specification (see Appendix F), spatial symmetries should not be used to decrease the number of plane waves. Note also that the dimension max\_number\_of\_coefficients actually governs the size of reduced\_coordinates\_of\_plane\_waves, so only when the gamma kpoint is present alone, will the size of the file effectively be reduced by the factor of two.

**coefficients\_of\_wavefunctions** Wavefunction coefficients. The wavefunctions must be normalized to 1, i.e. the sum of the absolute square of the coefficients of one wavefunction must be 1. See section 5.2. The attribute used\_time\_reversal\_at\_gamma must be used in the same way as for the variable

Variables	Type (index order as in C)	Notes
basis_set	char(character_string_length)	“plane_waves” if a plane-wave basis set is used “Daubechies_wavelets” if a Daubechies wavelet is used
number_of_coefficients	integer[number_of_kpoints]	The attribute “k_dependent” must be defined (see Table 15). Possible splitting, see Table 20.
coefficients_of_wavefunctions	double [number_of_spins] [number_of_kpoints] [max_number_of_states] [number_of_spinor_components] [max_number_of_coefficients] [real_or_complex_coefficients]	For both plane-wave basis set and Daubechies wavelet basis set. For wavelets, see comment (3) in section 5.2. Normalization for plane waves : 1 per unit cell, see section 5.2 See also possible modifications for split files in Table 20. The attribute used_time_reversal_at_gamma might be defined. Normalisation for wavelets should be defined in comment (3) of section 5.2.
Attributes	Type (index order as in C)	
used_time_reversal_at_gamma	char(80)	attribute of reduced_coordinates_of_plane_waves and coefficients_of_wavefunctions flag-type, see section 2.3

Table 16: Specification of wavefunctions in a basis set. Needed only in case “coefficients\_of\_wavefunctions” will be the array containing the wavefunctions.

reduced\_coordinates\_of\_plane\_waves .

**coordinates\_of\_basis\_grid\_points** For wavelets. Coordinates of the grid points where coefficients can be stored. This is used to define a real space basis set where a reduced set of points is used. This array may be used in conjunction with the variable number\_of\_coefficients\_per\_grid\_points.

**number\_of\_coefficients\_per\_grid\_points** For wavelets. This array gives the number of coefficients stored on basis set grid points. The coordinates of corresponding grid points are given in the array coordinates\_of\_basis\_grid\_points.

**order\_of\_Daubechies\_wavelets** For wavelets. This number gives the order of the Daubechies wavelet basis, e.g. if order\_of\_Daubechies\_wavelets is 14, the Daubechies wavelet are made from piecewise polynomial of order 14.

**used\_time\_reversal\_at\_gamma** Flag-type attribute (see section 2.3), that can be used for the variables reduced\_coordinates\_of\_plane\_waves and coefficients\_of\_wavefunctions

Variables	Type (index order as in C)	Notes
reduced_coordinates_of_plane_waves	integer[ number_of_kpoints] [max_number_of_coefficients] [number_of_reduced_dimensions]	The attribute “k_dependent” must be defined (see Table 15). See possible modifications for split files in Table 20. The attribute used_time_reversal_at_gamma might be defined.

Table 17: Specification of wavefunctions : additional data for a plane-wave basis.

Variables	Type (index order as in C)	Notes
coordinates_of_basis_grid_points	integer [number_of_localization_regions] [max_number_of_basis_grid_points] [number_of_reduced_dimensions]	For wavelets. See comment (3) in section 5.2.
number_of_coefficients_per_grid_point	integer [number_of_localization_regions] [max_number_of_basis_grid_points]	For wavelets. See comment (3) in section 5.2.
order_of_Daubechies_wavelets	integer	For wavelets. . See comment (3) in section 5.2.

Table 18: Specification of wavefunctions : additional data for a wavelet basis.

#### 5.1.4 Detailed description (table 19)

**real\_space\_wavefunctions** Wavefunction coefficients. Unlike for explicit basis set, the wavefunctions must be normalized to 1 per unit cell, i.e. the sum of the absolute square of the coefficients of one wavefunction, for all points in the grid, divided by the number of points must be 1. See section 5.2 . Note that this array has a number of dimensions that exceeds the maximum allowed in Fortran (that is, seven). This leads to practical problems only if the software to read/write this array attempts to read/write it in one shot. Our suggestion is instead to read/write sequentially parts of this array, e.g. to write the spin up part of it, and then, add the spin down. This might be done using Fortran arrays with at most seven dimensions.

Different variables see their dimensions modified, in case the file is split, as described in Section 2.4 (see Tables 5 and 7). In Table 20, we have gathered the variables whose dimensions will change. We have also dimensioned them as if the splitting was done on all the possible dimensions. This will rarely be the case, but intermediate situations can easily be deduced from the data gathered in this table.

Variables	Type (index order as in C)	Notes
real_space_wavefunctions	double [number_of_spins] [number_of_kpoints] [max_number_of_states] [number_of_spinor_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex_wavefunctions]	Normalization : 1 per unit cell. See possible modifications for split files in Table 20

Table 19: Specification of wavefunctions in real space.

The variables mentioned in Table 21 are optional. Note : they have been introduced in the present specification in prevision of use by some GW/BSE softwares, and might be subject to (heavy?) revisions in future versions of the specification.

### 5.1.5 Detailed description (table 21)

**gw\_corrections** GW-corrections to one-particle eigenvalues (see Table 20). Imaginary part (originating from the non-hermiticity) is optional. Should be 0 if unknown.

**max\_number\_of\_angular\_momenta** The maximum number of angular momenta to be considered for non-local Kleinman-Bylander separable norm-conserving pseudopotentials. If there is no non-local part, set it to 0. If the s channel is the highest angular momentum channel over all atomic species, then set it to 1. If the p channel (resp. d or f) is the highest, set it to 2 (resp. 3 or 4).

**max\_number\_of\_projectors** The maximum number of projectors for non-local Kleinman-Bylander separable norm-conserving pseudopotentials, over all angular momenta and all atomic species. If there is no non-local part, set it to 0. Most separable norm-conserving pseudopotentials have only one projector per angular momentum channel.

**kb\_formfactor\_sign** An array of integers whose value depend on the specific atomic species, angular momentum, and projector. It can have three values : when 0, it means that there is no projector defined for that channel. When +1 or -1, it gives the sign of the Kleinman-Bylander projector for that channel, as explained in section 5.2 .

**kb\_formfactors** and **kb\_formfactor\_derivatives** Kleinman-Bylander form factors in reciprocal space, and their derivative, as explained in section 5.2.

## 5.2 Comments

1. On the density, kpoint weights and occupation numbers. Supposing  $\rho_{n,k}(r)$  to be the partial density at point  $r$  (in real space, using reduced coordinates) due to band  $n$  at  $k$ -point  $k$  (in reciprocal space, using reduced

Variables	Type (index order as in C)	Notes
reduced_coordinates_of_kpoints	double[my_number_of_kpoints] [number_of_reduced_dimensions]	
number_of_coefficients	integer[my_number_of_kpoints]	
kpoint_weights	double[my_number_of_kpoints]	
occupations	double[my_number_of_spins] [my_number_of_kpoints] [my_max_number_of_states]	
eigenvalues	double[my_number_of_spins] [my_number_of_kpoints] [my_max_number_of_states]	Units attribute required. The attribute “scale_to_atomic_units” might also be mandatory, see section 2.2.
real_space_wavefunctions	double [my_number_of_spins] [my_number_of_kpoints] [my_max_number_of_states] [my_number_of_spinor_components] [my_number_of_grid_points_vector1] [my_number_of_grid_points_vector2] [my_number_of_grid_points_vector3] [real_or_complex_wavefunctions]	
coefficients_of_wavefunctions	double [my_number_of_spins] [my_number_of_kpoints] [my_max_number_of_states] [my_number_of_spinor_components] [my_max_number_of_coefficients] [real_or_complex_coefficients]	
reduced_coordinates_of_plane_waves	integer[my_number_of_kpoints] [number_of_reduced_dimensions]	

Table 20: Variables whose sizes are modified, in case of split files (assuming all splittings have been activated).

coordinates), then the full density at point is obtained thanks to

$$\rho(r^{red} \alpha) = \sum_{s \in sym} \sum_k w_k \sum_n f_{n,k} \rho_{n,k} (S_{s,\alpha\beta}^{red} (r_{\beta}^{red} - t_{s,\beta}^{red})) \quad (2)$$

where  $w_k$  is contained in the array “kpoint\_weights” of Table 14, and  $f_{n,k}$  is contained in the array “occupations” of Table 15 . This relation generalizes to the collinear spin-polarized case, as well as the non-collinear case by taking into account the “number\_of\_components” defined in Table 5 , and the direction of the magnetization vector.

2. On the Kleinman-Bylander form factors. One can always write the non-local part of Kleinman-Bylander pseudopotential (reciprocal space) in the

Dimensions	Type (index order as in C)	Notes
max_number_of_angular_momenta	integer	
max_number_of_projectors	integer	
Variables	Type (index order as in C)	Notes
gw_corrections	double[number_of_spins] [number_of_kpoints] [max_number_of_states] [real_or_complex_gw_corrections]	The “units” attribute is required. The attribute “scale_to_atomic_units” might also be mandatory, see section 2.2. Possibles changes for split files, as in Table 20
kb_formfactor_sign	integer[number_of_atom_species] [max_number_of_angular_momenta] [max_number_of_projectors]	
kb_formfactors	double[number_of_atom_species] [max_number_of_angular_momenta] [max_number_of_projectors] [number_of_kpoints] [max_number_of_coefficients]	Possible changes for split files, as in Table 20
kb_formfactor_derivative	double[number_of_atom_species] [max_number_of_angular_momenta] [max_number_of_projectors] [number_of_kpoints] [max_number_of_coefficients]	Possible changes for split files, as in Table 20

Table 21: Optional dimensions and variables that might be needed for some GW/BSE softwares

following way :

$$v_{nonloc}^{KB}(\vec{K}, \vec{K}') = \sum_s \left[ \sum_{a(s)} e^{-i(\vec{K}-\vec{K}')\vec{\tau}_a} \right] \left[ \sum_{lp} P_l(\hat{K} \cdot \hat{K}') F_{slp}^*(K) S_{slp} F_{slp}(K') \right] \quad (3)$$

with  $\vec{K} = \vec{k} + \vec{G}$ ,  $\vec{k}$  is one of the kpoints (see Table 13),  $\vec{G}$  is a vector of the reciprocal lattice, the list of reduced coordinates of which can be found in the variable reduced\_coordinates\_of\_plane\_waves of Table 16.  $K$  is the module of  $\vec{K}$  and  $\hat{K}$  its direction.  $\vec{\tau}_a$  is the atomic position of atom  $a$  belonging to species  $s$ .  $P_l(x)$  is the Legendre polynomial of order  $l$ .  $F_{slp}(K)$  is the Kleinman-Bylander form factor for species  $s$ , angular polynomial of order  $l$ , and number of projector  $p$ .  $S_{slp}$  is the sign of the dyadic product  $F_{slp}^*(K)F_{slp}(K')$ . The sum on  $a(s)$  runs over all atoms of atomic species  $s$ ,  $l$  runs over all the pseudopotential angular momentum components of the atomic species  $s$ , and  $p$  runs over the number of projectors allowed for a specific angular momentum channel of atomic species  $s$ . The additional variable kb\_formfactor\_derivative is equal to  $dF_{slp}(K)/dK$

3. On a Daubechies wavelet basis set with two levels of resolution. When the variable `basis_set` is set to “`daubechies_wavelets`”, the basis set is constituted by a reduced set of grid points that can host one or several coefficients. The following explanation assumes a two-level resolution but it can be used for other values. In the two-resolution case, all other quantities than the wavefunctions (as the density) are usually expressed on the finest grid, i.e. the grid for the density is twice the grid for the wavefunctions. Since dimensions `number_of_grid_points_vector[i]` are used to define the scalar variables, the `coordinates_of_basis_grid_points` must be even numbers in the two- resolution case. The wavefunctions are expanded in real space on a non-complete uniform grid. The grid points used for the basis set are listed in the variable `coordinates_of_basis_grid_points`. Each basis grid point can host one or eight coefficients as stored in the variable `number_of_coefficients_per_grid_points`. Then, in that case, the dimension `max_number_of_coefficients` is the sum over the basis gridpoint of the values of `number_of_coefficients_per_grid_point`. To build the wavefunctions from the values stored in `coefficients_of_wavefunctions`, one must read for each basis grid point the required number of coefficients. When one coefficient is given, this means a coefficient for a product of 1-dimensional Daubechies scaling-functions centered on the basis grid point. When eight values are given, this means eight coefficients for product of both scaling functions and wavelet functions ( $\phi$  denotes Daubechies scaling functions and  $\psi$  Daubechies wavelet functions) :

- $\phi(x)\phi(y)\phi(z)$
- $\psi(x)\phi(y)\phi(z)$
- $\phi(x)\psi(y)\phi(z)$
- $\psi(x)\psi(y)\phi(z)$
- $\phi(x)\phi(y)\psi(z)$
- $\psi(x)\phi(y)\psi(z)$
- $\phi(x)\psi(y)\psi(z)$
- $\psi(x)\psi(y)\psi(z)$

For a review on wavelets, including the description of Daubechies wavelets, see, e.g., *Wavelets and Their Application for the Solution of Partial Differential Equations in Physics*, Presses Polytechniques et Universitaires Romandes, Lausanne, (1998) by S. Goedecker.

### 5.3 Discussion and future

1. This specification should be extended to the PAW as well as LAPW methods. We suggest to supplement the present set of variables (so, using the same names) with other variables to be fixed later.

2. The specification of wavefunctions, Table 16 as well as 19, can accommodate the response wavefunctions of Density-Functional Perturbation Theory. On the contrary, the response eigenenergies (actually a hermitian matrix of Lagrange multipliers) cannot be accommodated by the “eigenvalues” array of Table 15.
3. Additional information has to be provided to deal with magnetic symmetry operations, in both collinear and non-collinear cases.

## 6 Dielectric function

### 6.1 Specification

A ETSF NetCDF file “containing a dielectric function” should contain at least the information needed to manipulate the resulting dielectric matrix or polarizability. To provide this and validation information as well, the file is required to be a valid “file containing crystallographic data” (Sect. 3). Of course, files might contain less information than that required here, but still follow the naming convention of ETSF. It might also contain more information, of the kind specified in other tables of the present document.

A ETSF NetCDF file “containing a dielectric function” should contain the following set of mandatory information, beyond the basic crystallographic data mentioned above :

1. The three attributes defined in Table 1
2. The following dimensions defined in Table 23
  - complex
  - number\_of\_reduced\_dimensions
  - number\_of\_frequencies\_dielectric\_function
  - number\_of\_qpoints\_dielectric\_function
  - number\_of\_qpoints\_gamma\_limit
  - number\_of\_spins
  - number\_of\_coefficients\_dielectric\_function
3. The following mandatory variables defined in Table 24
  - basis\_set
  - frequencies\_dielectric\_function
  - qpoints\_dielectric\_function
  - qpoints\_gamma\_limit
  - reduced\_coordinates\_of\_plane\_waves\_dielectric\_function
4. One of the following main content variables defined in Table 25
  - dielectric\_function
  - inverse\_dielectric\_function
  - polarizability
  - inverse\_polarizability
5. The optional attribute approximation\_scheme, associated to one of the variables in the preceding point.

6. The following mandatory head and wings variables defined in Table 26, and Table 28 depending on whether `dielectric_function` or `polarizability` is selected above

- `dielectric_function_head`
- `dielectric_function_upper_wing`
- `dielectric_function_lower_wing` (optional)
- `inverse_dielectric_function_head`
- `inverse_dielectric_function_upper_wing`
- `inverse_dielectric_function_lower_wing` (optional)
- `polarizability_head`
- `polarizability_upper_wing`
- `polarizability_lower_wing` (optional)
- `inverse_polarizability_head`
- `inverse_polarizability_upper_wing`
- `inverse_polarizability_lower_wing` (optional)

As mentioned in section 1 and 2, such a file might contain additional information, such as any of the variables specified in section 2. It might even contain enough information to be declared a ETSF NetCDF file “containing a wavefunction” or “containing the density”, and so on. Such a file might also contain additional information specific to the software that generated the file. It is not expected that this other software-specific information might be used by another software.

A ETSF NetCDF file of Dielectric Function type can contain several different quantities, calculated at several different levels of approximation, and specified through 1) the choice of the main variable being one of:

- `dielectric_function`
- `inverse_dielectric_function`
- `polarizability`
- `inverse_polarizability`

and 2) the modifier contained in the attribute `approximation_scheme`, which further specifies if, e.g., the polarizability is the non-interacting one or not, or if the Random Phase Approximation dielectric function is stored.

### 6.1.1 Detailed description (table 22)

**`approximation_scheme`** Attribute associated to one of the variables in Table 25. Describes the type of matrix which is stored in `polarizability`, `dielectric_function`, or their inverses, whichever is present. Possible values are the following:

Attribute	Type (index order as in C)	Notes
approximation_scheme	char(80)	

Table 22: The optional attribute for the approximation scheme of the polarizability or dielectric matrices.

- “independent particle”
- “random phase approximation”
- “interacting”
- “reducible”
- “irreducible”

NOTE: are any of these redundant? Some are mutually exclusive. They should perhaps be separated into several attributes.

Dimension	Type (index order as in C)	Notes
complex	integer	always equal to 2
number_of_reduced_dimensions	integer	always equal to 3 (could be changed for codes using fewer dimensions)
number_of_frequencies_dielectric_function	integer	
number_of_qpoints_dielectric_function	integer	
number_of_qpoints_gamma_limit	integer	
number_of_spins	integer	1 or 2
number_of_coefficients_dielectric_function	integer	No. of G-vectors or No. of real space points

Table 23: The specification of the dimensions for the dielectric function and related quantities

### 6.1.2 Detailed description (table 23)

**complex** The complex size of the dielectric arrays. Always set to 2. Normally response functions will always have an imaginary part.

**number\_of\_reduced\_dimensions** Always set to 3. May be modified in future to allow for calculations in different numbers of dimensions.

**number\_of\_frequencies\_dielectric\_function** Number of frequencies at which the dielectric function is calculated.

**number\_of\_qpoints\_dielectric\_function** Number of q-points in reciprocal space at which the dielectric function is calculated.

**number\_of\_qpoints\_gamma\_limit** Number of q-points at which the  $q \rightarrow 0$  limit is given for the head and wings of the dielectric matrix. See variable `qpoints.gamma.limit` below.

**number\_of\_spins** Number of spin channels for the dielectric matrix. There is no dielectric\_function suffix as it must be the same as for any other variable which is in the same file, such as densities or wavefunctions (whereas the number of coefficients can vary, for instance).

**number\_of\_coefficients\_dielectric\_function** Number of coefficients (in version 3.3, this is restricted to plane waves, i.e. reciprocal space  $G$  vectors) in the basis set for the dielectric matrix.

Variables	Type (index order as in C)	Notes
basis_set	char(character_string_length)	
frequencies_dielectric_function	double [number_of_frequencies_dielectric_function] [complex]	
qpoints_dielectric_function	double [number_of_qpoints_dielectric_function] [number_of_reduced_dimensions]	
qpoints_gamma_limit	double [number_of_qpoints_gamma_limit] [number_of_reduced_dimensions]	
reduced_coordinates_plane_waves_dielectric_function	integer [number_of_qpoints_dielectric_function] [number_of_coefficients_dielectric_function] [number_of_reduced_dimensions]	

Table 24: The specification of variables associated to the polarizability or dielectric function

### 6.1.3 Detailed description (table 24)

**basis\_set** The basis set used to represent the dielectric\_function. The primary intent is to use a plane wave basis, and reciprocal space vectors. “plane\_waves” if a plane-wave basis set is used. “real\_space” in case of real space formalism.

**frequencies\_dielectric\_function** The frequencies (real and/or imaginary) at which the dielectric function has been calculated. These are complex as they can be on the real or imaginary axis for different schemes.

**qpoints\_dielectric\_function** The q-points in reciprocal space at which the dielectric function has been calculated. In reduced coordinates of the reciprocal lattice vectors.

**qpoints\_gamma\_limit** The q-points in reciprocal space at which the head and wings of the dielectric function have been calculated (for the limit near the  $\Gamma$  point of the Brillouin Zone, ie long wavelengths). Two types of vector are possible:

1. Small but finite vectors at which the arrays dielectric\_function\_head and dielectric\_function\*\_wings have been evaluated. In reduced coordinates of the reciprocal lattice vectors.
2. If an analytic limit for  $q \rightarrow 0$  is present, e.g. calculated through k.p perturbation theory, the qpoints should be in the appropriate direction, but with norm 1.

**reduced\_coordinates\_plane\_waves\_dielectric\_function** Plane-wave G-vectors in reduced coordinates. Present only if the dielectric function/polarizability is described within the reciprocal space formalism i.e. `basis_set` is “plane\_waves”. The full flexibility of choosing different numbers of plane waves, or different spheres, at each q-point, is available.

Variables	Type (index order as in C)	Notes
dielectric_function	double [number_of_frequencies_dielectric_function] [number_of_qpoints_dielectric_function] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [number_of_coefficients_dielectric_function] [complex]	
inverse_dielectric_function	double [number_of_frequencies_dielectric_function] [number_of_qpoints_dielectric_function] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [number_of_coefficients_dielectric_function] [complex]	
polarizability	double [number_of_frequencies_dielectric_function] [number_of_qpoints_dielectric_function] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [number_of_coefficients_dielectric_function] [complex]	
inverse_polarizability	double [number_of_frequencies_dielectric_function] [number_of_qpoints_dielectric_function] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [number_of_coefficients_dielectric_function] [complex]	

Table 25: The specification of the possible dielectric-function-like matrices

#### 6.1.4 Detailed description (table 25)

All of the following variables have an associated attribute, `approximation_scheme`, described in Table 22.

**dielectric\_function** The dielectric function  $\epsilon = 1 - v\chi$ , where  $v$  is the Coulomb interaction and  $\chi$  is the interacting density-density response function, as a matrix in reciprocal space lattice vectors (called  $G$  and  $G'$ ).  $\epsilon$  is a function

of frequency (real or imaginary) and a qpoint in the irreducible Brillouin Zone.

*Remarks* The head and wings (for  $q=0$  and  $G=G'=0$ ) of this array must contain physically reasonable quantities, e.g. isotropic averages or estimates of the real head and wings, (which have non-analytic behaviors)

**inverse\_dielectric\_function** The inverse matrix of dielectric\_function. This implies full calculation of local field effects etc... It would not be useful to store the inverse without local field effects as it is an approximation, and is trivially obtained from the normal dielectric\_function.

*Remarks* The head and wings (for  $q=0$  and  $G=G'=0$ ) of this array must contain physically reasonable quantities, e.g. isotropic averages or estimates of the real head and wings, (which have non-analytic behaviors)

**polarizability** The irreducible polarizability ( $P$  or  $\chi_0$ ) =  $\delta n(r, t) / \delta v_{ext}(r', t')$ , Fourier transformed to reciprocal space and to frequency.  $P$  is a function of (real or complex) frequency, qpoint, and as a matrix in  $G$  space. Other basis sets can be accomodated for by specifying their number and type of coefficients.

*Remarks* The head and wings (for  $q=0$  and  $G=G'=0$ ) of this array must contain physically reasonable quantities, e.g. isotropic averages or estimates of the real head and wings, (which have non-analytic behaviors)

**inverse\_polarizability** The inverse matrix of polarizability. This implies full calculation of local-field-like effects etc... The inverse polarizability is sometimes needed to include vertex-corrections in the calculation of the dielectric function.

*Remarks* The head and wings (for  $q=0$  and  $G=G'=0$ ) of this array must contain physically reasonable quantities, e.g. isotropic averages or estimates of the real head and wings, (which have non-analytic behaviors).

### 6.1.5 Detailed description (table 26)

**dielectric\_function\_head** The head ( $G = 0, G' = 0$  element) of the dielectric function. The value will depend on the way the  $q \rightarrow 0$  limit is taken, and different values are stored according to:

1. Either the small but finite qpoints which are used to evaluate the head
2. The directions in which the derivative is taken, if analytic formulae are employed.

$$\text{dielectric\_function\_head}(\omega) = \lim_{q \rightarrow 0} \epsilon(\vec{G} = 0, \vec{G}' = 0, \vec{q}, \omega) \quad (4)$$

*Remarks*

**dielectric\_function\_upper\_wing**

**dielectric\_function\_lower\_wing**

The wings of the dielectric function are the elements of the dielectric\_function matrix for  $q \rightarrow 0$  and for either  $G = 0$  and  $G' \neq 0$  or for  $G' = 0$  and  $G \neq 0$ . As for the head, the value depends on the limiting direction, and the same qpoint specifications are used.

Variables	Type (index order as in C)	Notes
dielectric_function_head	double [number_of_frequencies_dielectric_function] [number_of_qpoints_dielectric_function] [number_of_spins] [number_of_spins] [complex]	
dielectric_function_upper_wing	double [number_of_frequencies_dielectric_function] [number_of_qpoints_gamma_limit] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [complex]	
dielectric_function_lower_wing	double [number_of_frequencies_dielectric_function] [number_of_qpoints_gamma_limit] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [complex]	

Table 26: The specification of the head and wings of the dielectric function

$$\text{dielectric\_function\_upper\_wing}(\omega) = \lim_{q \rightarrow 0} \epsilon(\vec{G} = 0, \vec{G}' \neq 0, \vec{q}, \omega) \quad (5)$$

$$\text{dielectric\_function\_lower\_wing}(\omega) = \lim_{q \rightarrow 0} \epsilon(\vec{G} \neq 0, \vec{G}' = 0, \vec{q}, \omega) \quad (6)$$

Usually both wings do not need to be specified, due to spatial symmetry of the response functions (invariance on exchanging the two real or reciprocal space arguments). The upper wing *must* be specified, the lower wing is optional.

*Remarks*

### 6.1.6 Detailed description (table 27)

**inverse\_dielectric\_function\_head** The head ( $G = 0, G' = 0$  element) of the inverse of the dielectric function. The value will depend on the way the  $q \rightarrow 0$  limit is taken, and different values are stored according to:

1. Either the small but finite qpoints which are used to evaluate the head
2. The directions in which the derivative is taken, if analytic formulae are employed.

$$\text{dielectric\_function\_head}(\omega) = \lim_{q \rightarrow 0} \epsilon^{-1}(\vec{G} = 0, \vec{G}' = 0, \vec{q}, \omega) \quad (7)$$

*Remarks*

Variables	Type (index order as in C)	Notes
inverse_dielectric_function_head	double [number_of_frequencies_dielectric_function] [number_of_qpoints_dielectric_function] [number_of_spins] [number_of_spins] [complex]	
inverse_dielectric_function_upper_wing	double [number_of_frequencies_dielectric_function] [number_of_qpoints_gamma_limit] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [complex]	
inverse_dielectric_function_lower_wing	double [number_of_frequencies_dielectric_function] [number_of_qpoints_gamma_limit] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [complex]	

Table 27: The specification of the head and wings of the inverse of the dielectric function

#### **inverse\_dielectric\_function\_upper\_wing**

#### **inverse\_dielectric\_function\_lower\_wing**

The wings of the inverse of the dielectric function are the elements of the inverse\_dielectric\_function matrix for  $q \rightarrow 0$  and for either  $G = 0$  and  $G' \neq 0$  or for  $G' = 0$  and  $G \neq 0$ . As for the head, the value depends on the limiting direction, and the same qpoint specifications are used.

$$\text{inverse\_dielectric\_function\_upper\_wing}(\omega) = \lim_{q \rightarrow 0} \epsilon^{-1}(\vec{G} = 0, \vec{G}' \neq 0, \vec{q}, \omega) \quad (8)$$

$$\text{inverse\_dielectric\_function\_lower\_wing}(\omega) = \lim_{q \rightarrow 0} \epsilon^{-1}(\vec{G} \neq 0, \vec{G}' = 0, \vec{q}, \omega) \quad (9)$$

Usually both wings do not need to be specified, due to spatial symmetry of the response functions (invariance on exchanging the two real or reciprocal space arguments). The upper wing *must* be specified, the lower wing is optional.

*Remarks*

### **6.1.7 Detailed description (table 28)**

#### **polarizability\_head**

The  $G = 0$  and  $G' = 0$  element of the polarizability matrix, in the limit  $q \rightarrow 0$ . The value of the head depends on the direction taken in the limit ( $\chi$  is a non-analytical function), specified by qpoints\_gamma\_limit, described above

Variables	Type (index order as in C)	Notes
polarizability_head	double [number_of_frequencies_dielectric_function] [number_of_qpoints_dielectric_function] [number_of_spins] [number_of_spins] [complex]	
polarizability_upper_wing	double [number_of_frequencies_dielectric_function] [number_of_qpoints_gamma_limit] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [complex]	
polarizability_lower_wing	double [number_of_frequencies_dielectric_function] [number_of_qpoints_gamma_limit] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [complex]	

Table 28: The specification of the head and wings of the polarizability

*Remarks*

**polarizability\_upper\_wing**

**polarizability\_lower\_wing**

The wings of the polarizability are the elements of the matrix for  $q \rightarrow 0$  and for either  $G = 0$  and  $G' \neq 0$  or for  $G' = 0$  and  $G \neq 0$ . As for the head, the value depends on the limiting direction, and the same qpoint specifications are used.

$$\text{polarizability\_upper\_wing}(\omega) = \lim_{q \rightarrow 0} \chi(\vec{G} = 0, \vec{G}' \neq 0, \vec{q}, \omega) \quad (10)$$

$$\text{polarizability\_lower\_wing}(\omega) = \lim_{q \rightarrow 0} \chi(\vec{G} \neq 0, \vec{G}' = 0, \vec{q}, \omega) \quad (11)$$

Usually both wings do not need to be specified, due to spatial symmetry of the response functions (invariance on exchanging the two real or reciprocal space arguments). The upper wing *must* be specified, the lower wing is optional.

### 6.1.8 Detailed description (table 29)

**inverse\_polarizability\_head**

The  $G = 0$  and  $G' = 0$  element of the inverse of the polarizability matrix, in the limit  $q \rightarrow 0$ . The value of the head depends on the direction taken in the limit ( $\chi^{-1}$  is a non-analytical function), specified by `qpoints.gamma.limit`, described above.

*Remarks*

**inverse\_polarizability\_upper\_wing**

Variables	Type (index order as in C)	Notes
inverse_polarizability_head	double [number_of_frequencies_dielectric_function] [number_of_qpoints_dielectric_function] [number_of_spins] [number_of_spins] [complex]	
inverse_polarizability_upper_wing	double [number_of_frequencies_dielectric_function] [number_of_qpoints_gamma_limit] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [complex]	
inverse_polarizability_lower_wing	double [number_of_frequencies_dielectric_function] [number_of_qpoints_gamma_limit] [number_of_spins] [number_of_spins] [number_of_coefficients_dielectric_function] [complex]	

Table 29: The specification of the head and wings of the inverse of the polarizability

### **inverse\_polarizability\_lower\_wing**

The wings of the inverse of the polarizability matrix are the elements of the matrix for  $q \rightarrow 0$  and for either  $G = 0$  and  $G' \neq 0$  or for  $G' = 0$  and  $G \neq 0$ . As for the head, the value depends on the limiting direction, and the same qpoint specifications are used.

$$\text{polarizability\_upper\_wing}(\omega) = \lim_{q \rightarrow 0} \chi^{-1}(\vec{G} = 0, \vec{G}' \neq 0, \vec{q}, \omega) \quad (12)$$

$$\text{polarizability\_lower\_wing}(\omega) = \lim_{q \rightarrow 0} \chi^{-1}(\vec{G} \neq 0, \vec{G}' = 0, \vec{q}, \omega) \quad (13)$$

Usually both wings do not need to be specified, due to spatial symmetry of the response functions (invariance on exchanging the two real or reciprocal space arguments). The upper wing *must* be specified, the lower wing is optional.

## **6.2 Comments**

## 7 Pseudopotential / PAW set up files

At the Louvain-la-Neuve meeting, three formats based on XML have been reported, as well as other text formats. A combined strategy has been decided: writing a conversion utility to cope with existing files (this is the short-term action); then writing routines to facilitate the use of XML-based formats, in view of hoped unification (this is the long-term action).

The different formats can be found on the Web site:

[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format](http://www.pcpm.ucl.ac.be/workshop_files/file_format)

In particular, note :

1. the XML format, for PAW set up files (a reduced set of these data is able to specify ultra- soft as well as norm-conserving pseudopotentials) from Jens Mortensen :  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format/psps-mortensen\\_1.ps](http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-mortensen_1.ps)  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format/psps-mortensen\\_2.ps](http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-mortensen_2.ps)
2. the XML format, for norm-conserving pseudopotential files, from J. Junquera and A. Garcia :  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format/psps-xml-garcia.txt](http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-xml-garcia.txt)  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format/psps-xml-garcia.tgz](http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-xml-garcia.tgz)  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format/psps-xml-junquera\\_v2.ppt](http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-xml-junquera_v2.ppt)  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format/psps-xml-junquera.txt](http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-xml-junquera.txt)
3. the “XML-like” UPF format presented by P. Gianozzi, used in the ESPRESSO package:  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format/psps-upf-gianozzi.txt](http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-upf-gianozzi.txt)  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format/psps-upf-gianozzi.f90](http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-upf-gianozzi.f90)

## 8 Files with other contents relevant to ETSF

Files containing the following information are also relevant to the ETSF activities:

- dielectric matrices (frequency-dependent), susceptibility matrices, screening matrices (typically for GW calculations)
- derivatives of the total energy, typically obtained from Density-Functional Perturbation Theory (dynamical matrices, Born effective charges, elastic constants, piezoelectric constants, etc ...).
- electron-phonon coupling elements Some of the existing file formats to contain these data were presented at the Louvain-la-Neuve meeting. The different formats can be found on the Web site:  
[http://www.pcpm.ucl.ac.be/workshop\\_files/file\\_format](http://www.pcpm.ucl.ac.be/workshop_files/file_format)

## 9 Appendix A: Some information on the NetCDF size limitation

(The following information has been provided shortly after the workshop by Valerio Olevano) To summarize :

1. The 2GB limit is firstly a FILE-SIZE limit of operating systems on 32-bits machine (and some non-updated 64-bits old-operating-systems). And this cannot be overcome, even splitting wavefunctions into  $n_{bands} * n_{kpoints}$  variables.
2. Assuming your machine can store  $\gg 2$ GB files, the NetCDF has in general a limit of 4GB. BUT even with the actual version you can store in NetCDF at least one variable (the last) up to Terabytes, and probably in future this will be extended to also the non last variables.

More detailed discussion :

### 4.4 NetCDF 64-bit Offset Format Limitations

Although the 64-bit offset format allows the creation of much larger NetCDF files than was possible with the classic format, there are still some restrictions on the size of variables. It is important to note that without Large File Support (LFS) in the operating system, it is impossible to create any file larger than 2 GBytes. Assuming an operating system with LFS, the following restrictions apply to the NetCDF 64-bit offset format.

No fixed-size variable can require more than  $2^{32} - 4$  bytes (i.e. 4GB - 4 bytes, or 4,294,967,292 bytes) of storage for its data, unless it is the last fixed-size variable and there are no record variables. When there are no record variables, the last fixed-size variable can be any size supported by the file system, e.g. terabytes.

A 64-bit offset format NetCDF file can have up to  $2^{32} - 1$  fixed sized variables, each under 4GB in size. If there are no record variables in the file the last fixed variable can be any size. No record variable can require more than  $2^{32} - 4$  bytes of storage for each record's worth of data, unless it is the last record variable. A 64-bit offset format NetCDF file can have up to  $2^{32} - 1$  records, of up to  $2^{32} - 1$  variables, as long as the size of one record's data for each record variable except the last is less than 4 GB - 4.

Note also that all NetCDF variables and records are padded to 4-byte boundaries.

## 10 Appendix B: List of changes of this version with respect to v2.2 of Jun 4, 2007

Changes from v2.2 to v3.0 :

Major modification :

pp 9-10 : added two new dimensions `max_number_of_basis_grid_points`, and `number_of_localization_regions`.

pp 22-23 : modified (4) and (9) for wavelet basis specification

pp 25-27 : added three new variables : `coordinates_of_basis_grid_points` and `number_of_coefficients_per_grid_point`, `order_of_Daubechies_wavelets` ; also split former table 16 into tables 16, 17, and 18 , and renumbered former tables 17, 18 and 19.

p 31 : full description of the wavelet basis set

pp 39-40 : updated list

Correction:

p.9 : table 5 had `my_number_of_spin_components` instead of `my_number_of_spinor_components`

p. 22 : number of `spinor_components` moved from (2) to (3)

## 11 Appendix C: List of things under debate

- Should formulate specification for other susceptibilities (spin, frequency, real or reciprocal representation), electron-phonon, dynamical matrices
- Tolerances / treshold for equality of two double numbers (e.g. k points, when given explicitly) ; one might define a tolerance in the specif, or define some tolerance variables ?!
- Specification should be clarified about Monkhorst-Pack sampling in case where the original article refers to conventional reciprocal lattice, and not primitive one
- Should discuss symmetries in case of magnetization (collinear and non-collinear).
- Should plan PAW / USPP generalisation , perhaps LAPW ?
- Should debate about the interest of a pseudopotential specif - perhaps PAW/USPP atomic data.

## **12 Appendix D. List of ETSF NetCDF agreed names (variable, attributes, dimensions, following alphabetical ordering)**

Note : all the variables/dimensions beginning with “my\_” refer to split files, and are explained in Tables 6 and Table 7 At present, there are 78 agreed variables/dimensions/attributes :

atom_species	Variable	Table 11
atom_species_names	Variable	Table 11
atomic_numbers	Variable	Table 11
basis_set	Variable	Table 16
character_string_length	Dimension	Table 4
chemical_symbols	Variable	Table 11
coefficients_of_wavefunctions	Variable	Table 16
Conventions	Global attribute	Table 1
coordinates_of_basis_grid_points	Variable	Table 16
correlation_functional	Variable	Table 9
correlation_potential	Variable	Table 13
density	Variable	Table 12
eigenvalues	Variable	Table 15
exchange_correlation_potential	Variable	Table 13
exchange_functional	Variable	Table 9
exchange_potential	Variable	Table 13
fermi_energy	Variable	Table 9
file_format	Global attribute	Table 1
file_format_version	Global attribute	Table 1
gw_corrections	Variable	Table 21
history	Global attribute	Table 2
k_dependent	Attribute	Table 15
kb_formfactor_sign	Variable	Table 21
kb_formfactors	Variable	Table 21
kb_formfactor_derivative	Variable	Table 21
kinetic_energy_cutoff	Variable	Table 10
kpoint_grid_shift	Variable	Table 10
kpoint_grid_vectors	Variable	Table 10
kpoints_weights	Variable	Table 14
max_number_of_angular_momenta	Dimension	Table 21
max_number_of_basis_grid_points	Dimension	Table 5
max_number_of_coefficients	Dimension	Table 5
max_number_of_projectors	Dimension	Table 21
max_number_of_states	Dimension	Table 5
monkhorst_pack_folding	Variable	Table 10
number_of_atoms	Dimension	Table 4
number_of_atom_species	Dimension	Table 4
number_of_cartesian_directions	Dimension	Table 4
number_of_coefficients	Variable	Table 16
number_of_coefficients_per_grid_point	Variable	Table 16
number_of_components	Dimension <sup>50</sup>	Table 5
number_of_electrons	Variable	Table 9
number_of_grid_points_vector1	Dimension	Table 5
number_of_grid_points_vector2	Dimension	Table 5
number_of_grid_points_vector3	Dimension	Table 5

number_of_kpoints	Dimension	Table 5
number_of_localization_regions	Dimension	Table 5
number_of_reduced_dimensions	Dimension	Table 4
number_of_spinor_components	Dimension	Table 5
number_of_spins	Dimension	Table 5
number_of_states	Variable	Table 15
number_of_symmetry_operations	Dimension	Table 4
number_of_vectors	Dimension	Table 4
occupations	Variable	Table 15
order_of_Daubechies_wavelets	Variable	Table 18
primitive_vectors	Variable	Table 11
pseudopotential_types	Variable	Table 8
real_or_complex_coefficients	Variable	Table 4
real_or_complex_density	Variable	Table 4
real_or_complex_gw_corrections	Variable	Table 4
real_or_complex_potential	Variable	Table 4
real_or_complex_wavefunctions	Variable	Table 4
real_space_wavefunctions	Variable	Table 19
reduced_atom_positions	Variable	Table 11
reduced_coordinates_of_kpoints	Variable	Table 14
reduced_coordinates_of_plane_waves	Variable	Table 16
reduced_symmetry_matrices	Variable	Table 11
reduced_symmetry_translations	Variable	Table 11
scale_to_atomic_units	Attribute	Table 3
smearing_scheme	Variable	Table 9
smearing_width	Variable	Table 9
space_group	Variable	Table 11
symbol_length	Dimension	Table 4
symmorphic	Attribute	Table 11
title	Global attribute	Table 2
units	Attribute	Table 3
used_time_reversal_at_gamma	Attribute	Table 16
valence_charges	Variable	Table 8