

# USER'S GUIDE

## DENCHAR 2.1

- *A program to plot charge densities and wave functions in real space* -

*May 4, 2012*

Javier Junquera

*Departamento de Ciencias de la Tierra y Física de la Materia Condensada  
Universidad de Cantabria, Santander, E-39005, Spain*

**javier.junquera@unican.es**

Pablo Ordejón, Alberto García

*Institut de Ciència de Materials de Barcelona - CSIC  
Campus de la UAB, 08193 Bellaterra, Barcelona, Spain*

**ordejon@icmab.es, albertog@icmab.es**

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>2</b>
<b>2</b>	<b>HOW TO COMPILE AND RUN THE PROGRAM</b>	<b>2</b>
2.1	Compilation . . . . .	2
2.2	Running the code . . . . .	3
<b>3</b>	<b>INPUT DATA FILE</b>	<b>4</b>
3.1	General descriptors . . . . .	4
3.2	Description of the 2D plane or 3D reference system . . . . .	5
<b>4</b>	<b>OUTPUT FILES</b>	<b>8</b>
4.1	2D mode . . . . .	8
4.1.1	Charge Density . . . . .	8
4.1.2	Wavefunctions . . . . .	9
4.2	3D mode . . . . .	9
4.2.1	Charge Density . . . . .	9
4.2.2	Wavefunctions . . . . .	10
<b>5</b>	<b>EXAMPLES</b>	<b>10</b>

# 1 INTRODUCTION

The program DENCHAR calculates charge densities and/or electronic wavefunctions in real space, from the output generated by SIESTA.

The code has two different modes of usage:

- **2D mode**, where the charge density and/or electronic wavefunctions are printed on a regular grid of points contained in a 2D plane (specified by the user). The results are printed in a file, in the form of a list  $x_i, y_i, f(x_i, y_i)$ , where  $f$  is either a charge density or a wavefunction, and  $x_i, y_i$  are the coordinates of each grid point in the plane, referred to a cartesian system where the  $x$  and  $y$  axes are contained in the plane and  $z$  is perpendicular to it. This file can be used to plot contour maps by means of 2D graphics packages.
- **3D mode**, where the charge density and/or electronic wavefunctions are printed on a regular grid of points in 3D. The results are printed in a file in Gaussian Cube format, that can be visualized by means of standard programs such as MOLDEL and MOLEKEL. The system of coordinates  $(x, y, z)$  used to produce the grid is orthonormal, and does not need to be the same as the one of SIESTA, but can be defined by the user.

The main features of DENCHAR are:

- Flexibility for defining the plane (in 2D mode) or the coordinates system (in 3D mode) where the charge density and/or wavefunctions will be computed. Possible options are: the normal vector, two lines contained in the plane, three points included in the plane, or three atoms contained in it. The input variables used to define the plane in 2D and the coordinates system in 3D are the same. In the latter case, the  $x$  and  $y$  axes are those defining the plane, and  $z$  is the perpendicular axis to them.
- For spin-unpolarized calculations, the fully self-consistent charge and the difference between the SCF charge density and the superposition of atomic densities are calculated. In this way, we can study how the chemical bonds change the charge distribution.
- For spin-polarized calculations, the magnetization (difference between the charge density for spin up and down) is calculated.
- When wavefunctions are computed, each wavefunction present in the file produced by SIESTA is printed on a different file, so that they can be plotted individually. Real and imaginary parts are written independently.

## 2 HOW TO COMPILE AND RUN THE PROGRAM

### 2.1 Compilation

The code for DENCHAR is now in the top source directory. Just type `make denchar` there to compile it.

## 2.2 Running the code

DENCHAR needs some information that will be supplied by SIESTA. The first thing we must do is to run SIESTA for the system we are interested in, setting up variable `WriteDenchar` to true. In this way SIESTA will generate two files called *SystemLabel*.PLD and *SystemLabel*.DIM where information needed to plot the charge density and/or wavefunctions in real space is dumped. You also need to have the files which contain the information about the basis set for each species in your system: *ChemicalSpecies*.ion (one for each chemical species), which are also generated by SIESTA. Finally, you will need the file containing the density matrix *SystemLabel*.DM if you wish to plot the charge density, or that containing the wavefunction coefficients *SystemLabel*.WFSX if you want to plot wavefunctions. These files need to be present in the directory where you are running DENCHAR (let's call it `~/denchar-run`).

Go into the directory where DENCHAR will run:

```
# cd ~/denchar-run
```

Copy all the required files into this directory; if you have run SIESTA in directory `~/siesta/RUN`:

```
# cp ~/siesta/RUN/SystemLabel.PLD .
```

```
# cp ~/siesta/RUN/SystemLabel.DIM .
```

```
# cp ~/siesta/RUN/*.ion .
```

```
# cp ~/siesta/RUN/SystemLabel.DM .
```

```
# cp ~/siesta/RUN/SystemLabel.WFSX .
```

(NOTE: Modern versions of SIESTA produce different kinds of WFSX files:

- *SystemLabel.fullBZ.WFSX* : for full PDOS or COOP/COHP analysis.
- *SystemLabel.bands.WFSX* : for “fatbands” analysis.
- *SystemLabel.selected.WFSX* : holding selected k-point information.

Be sure to copy or link the right version to *SystemLabel.WFSX*.)

Copy or link the executable file to the directory where you intend to run DENCHAR, and that contains the files described above:

```
# ln -s ~/siesta/Util/Denchar/Src/denchar .
```

Edit an input file for DENCHAR, *input.fdf*, following the instructions given in Section 3. Then, run DENCHAR reading *input.fdf* on standard input:

```
# denchar < input.fdf
```

The output with the values of charges and wavefunctions in the grid is dumped into files, which will be described in Section 4. Some informative output is also given on standard output.

### 3 INPUT DATA FILE

The input data file is written in an special format called FDF, developed by Alberto García and José M. Soler (see SIESTA USER'S GUIDE). It contains all the parameters needed to specify the details of the run and to define the plane or coordinates system. Here is a description of the variables that you can define in your DENCHAR input files, with their data types and default values.

#### 3.1 General descriptors

**SystemLabel** (*string*): A **single** word (max. 20 characters **without blanks**) containing a nickname of the system, used to name output files. **It must be the same that you use when you run SIESTA!!**

*Default value:* siesta

**NumberOfSpecies** (*integer*): Number of different atomic species in the simulation. Atoms of the same species, but with a different pseudopotential or basis set are counted as different species. Use the same value as in the SIESTA run.

*Default value:* There is no default. You must supply this variable.

**ChemicalSpeciesLabel** (*data block*): It specifies the different chemical species that are present, assigning them a number for further identification. Use the same as you did in SIESTA.

*Use:* This block is mandatory.

*Default:* No default.

**Denchar.TypeOfRun** (*string*): Specifies if you want to represent your charge density and/or wavefunctions in 2D or 3D.

*Default value:* 2D

**Denchar.PlotCharge** (*logical*):

Specifies if you want to calculate and plot a charge density in real space.

*Use:* Either **Denchar.PlotCharge** or **Denchar.PlotWaveFunctions** must be **.TRUE.**. If set to **.TRUE.**, file *SystemName.DM* should be present.

*Default value:* **.FALSE.**

**Denchar.PlotWaveFunctions** (*logical*):

Specifies if you want to calculate and plot wavefunctions in real space.

*Use:* Either **Denchar.PlotCharge** or **Denchar.PlotWaveFunctions** must be **.TRUE.**. If set to **.TRUE.**, file *SystemName.WFSX* should be present.

*Default value:* **.FALSE.**

## 3.2 Description of the 2D plane or 3D reference system

**Denchar.CoorUnits** (*string*): Character string to specify the format of the position of the points that define the  $xy$  plane in input. These can be expressed in two forms:

- Ang : Angstroms
- Bohr : Bohrs

*Default value:* Bohr

**Denchar.DensityUnits** (*string*): Character string to specify the units of the charge density in output. These can be expressed in three forms:

- Ele/Bohr\*\*3 : Electrons/bohr\*\*3
- Ele/Ang\*\*3 : Electrons/angstrom\*\*3
- Ele/UnitCell : Electrons/Unit Cell

*Default value:* Ele/bohr\*\*3

**Denchar.NumberPointsX** (*integer*): Number of subdivision of the grid in the x-direction. Together with **Denchar.NumberPointsY** (and **Denchar.NumberPointsZ** for 3D mode), it will define the number of points of the grid to plot the charge density and/or wavefunctions.

*Default value:* 50

**Denchar.NumberPointsY** (*integer*): Number of subdivision of the grid in the y-direction. Together with **Denchar.NumberPointsX** (and **Denchar.NumberPointsZ** for 3D mode), it will define the number of points of the grid to plot the charge density and/or wavefunctions.

*Default value:* 50

**Denchar.NumberPointsZ** (*integer*): Number of subdivision of the grid in the z-direction. Together with **Denchar.NumberPointsX** and **Denchar.NumberPointsY** it defines the number of points of the grid to plot the the charge density and/or wavefunctions.

*Use:* Only used if **Denchar.TypeOfRun** = 3D

*Default value:* 50

The next four variables define the size of the window inside the plane where we will focus our attention.

**Denchar.MinX** (*real length*): Defines the minimum value of the x-component of the 2D or 3D grid, in the system or reference of the plotting plane (for 2D) or the 3D grid axes (in 3D).

*Default value:* -3.0 bohrs

**Denchar.MaxX** (*real length*): Defines the maximum value of the x-component of the 2D or 3D grid, in the system or reference of the plotting plane (for 2D) or the 3D grid axes (in 3D).

*Default value:* +3.0 bohrs

**Denchar.MinY** (*real length*): Defines the minimum value of the y-component of the 2D or 3D grid, in the system or reference of the plotting plane (for 2D) or the 3D grid axes (in 3D).

*Default value:* -3.0 bohrs

**Denchar.MaxY** (*real length*): Defines the maximum value of the y-component of the 2D or 3D grid, in the system or reference of the plotting plane (for 2D) or the 3D grid axes (in 3D).

*Default value:* +3.0 bohrs

**Denchar.MinZ** (*real length*): Defines the minimum value of the z-component of the 3D grid, in the system or reference of the the 3D grid axes.

*Use:* Only used if **Denchar.TypeOfRun** = 3D

*Default value:* -3.0 bohrs

**Denchar.MaxZ** (*real length*): Defines the maximum value of the z-component of the 3D grid, in the system or reference of the the 3D grid axes.

*Use:* Only used if **Denchar.TypeOfRun** = 3D

*Default value:* +3.0 bohrs

**Denchar.PlaneGeneration** (*string*): Select the option to generate the *xy* plane (which is the plane of the plot in 2D mode, and that defines the *x-y* axes in 3D mode).

- NormalVector : If we want to choose the normal vector to describe the plane.
- TwoLines : If we want to specify two vectors contained in the plane.
- ThreePoints : If we want to give the coordinates of three points of the plane.
- ThreeAtomicIndices : If we want the plane that contains three given atoms.

*Default value:* NormalVector

**Denchar.CompNormalVector** (*data block*): Components of the normal vector. A normal vector defines a family of parallel planes. So we must specify which of these planes is the one we are interested in. So, when we select the option *NormalVector* to generate the plane, we must also input the origin of our plane (one point lying on it), and another point to define the x direction inside the plane (see below).

*Use:* Used only if **Denchar.PlaneGeneration** is NormalVector

*Default value:*

0.000 0.000 1.000

**Denchar.Comp2Vectors** (*data block*): Components of two vectors contained in the plane. The first vector defines the x direction inside the plane. If **Denchar.PlaneGeneration**=*TwoLines* we must also supply the origin of the plane, *i.e.* the coordinates of one point inside the plane (see **Denchar.PlaneOrigin** below).

*Use:* Used only if **Denchar.PlaneGeneration** is *TwoLines*

*Default value:*

```
1.000  0.000  0.000
0.000  1.000  0.000
```

**Denchar.Coor3Points** (*data block*): Coordinates of three points inside the plane. The first one will be taken as the origin of the plane. The vector between the first and the second one will determine the x-direction inside the plane.

*Use:* Used only if **Denchar.PlaneGeneration** is *ThreePoints*

*Default value:*

```
1.000  0.000  0.000
0.000  1.000  0.000
0.000  0.000  1.000
```

**Denchar.Indices3Atoms** (*data block*): Indices of three atoms that will belong to the plane. In this way, we define the plane that contains three given atoms. The coordinates of the first atom are taken as the origin of the plane, and the vector between the first and second atom will define the x-direction within the plane.

*Use:* Used only if **Denchar.PlaneGeneration** is *ThreeAtomicIndices*

*Default value:*

```
1 2 3
```

**Denchar.PlaneOrigin** (*data block*): Coordinates of one point inside the plane that will be taken as the origin. This is necessary if we want to define the plane from the normal vector or from two lines, because we must select one of the planes of the family of parallel planes.

*Use:* Used only if **Denchar.PlaneGeneration** is *NormalVector* or *TwoLines*. If **Denchar.PlaneGeneration** is *ThreePoints* or *ThreeAtomicIndices*, the **Denchar.PlaneOrigin** is automatically chosen (see description of variables **Denchar.Coor3Points** and **Denchar.Indices3Atoms**).

*Default value:*

```
0.000  0.000  0.000
```

**Denchar.X-Axis** (*data block*): Coordinates of one point inside the plane needed to define the x-direction, when the normal vector is selected to define the plane. The vector between the origin and this new point will define the x-direction.



*Units:* bohrs.

*Use:* Used only if **Denchar.PlaneGeneration** is **NormalVector**

*Default value:*

1.000 0.000 0.000

**Denchar.AtomsInPlane** (*data block*): Indices of the atoms whose coordinates will be rotated to the in-plane reference frame. In this system of reference, atoms in the plane will have the third coordinate equal to zero. The coordinates will be written in the corresponding output files. The units of the output coordinates will be determined by **Denchar.CoorUnits**. One index per line.

*Default value:* No default value

## 4 OUTPUT FILES

The output files produced by DENCHAR depend on the mode of run (2D or 3D) and the quantities being plotted (charge density and/or wavefunctions).

### 4.1 2D mode

The output files generated in 2D mode runs have all the same format: three columns, with the first two columns giving the coordinates of the points in the plane (in the reference frame in which  $x$  and  $y$  are contained in the plane), and the third column gives the corresponding charge density or wavefunction at that point. These can be used to draw contour maps or other 2D graphic representations by means of standard graphics programs. Depending on the physical quantity being plotted, different files are generated:

#### 4.1.1 Charge Density

If the calculation is not spin-polarized, two output files will be generated by DENCHAR:

*SystemLabel.CON.SCF* : Self-Consistent Charge Density at the points of the plane in real space.

*SystemLabel.CON.DEL* : Difference between self-consistent charge density and the superposition of atomic densities.

If the calculation is spin-polarized, there are four output files, namely:

*SystemLabel.CON.UP* : Self-consistent charge density for electrons with spin UP.

*SystemLabel.CON.DOWN* : Self-consistent charge density for electrons with spin DOWN.

*SystemLabel.CON.MAG* : **Magnetization**. Difference between self-consistent charge density with spin up and spin down.

*SystemLabel.CON.DEL* : Difference between self-consistent charge density (equal to the sum of charge density for both spines) and the superposition of atomic densities.

#### 4.1.2 Wavefunctions

DENCHAR plots each of the wavefunctions that is present in the file *SystemLabel.WFSX* generated by Siesta. For each wavefunction, a different file will be created containing the values of that wavefunction. In what follows, the wave function number is denoted by #.

If the calculation is not spin-polarized, one output file will be generated by DENCHAR for each wavefunction:

*SystemLabel.CON.WF#* : Values of the wave function number #

If the calculation is spin-polarized, there are two output files, namely:

*SystemLabel.CON.WF#.UP* : Values of the wave function number # with spin UP.

*SystemLabel.CON.WF#.DOWN* : Values of the wave function number # with spin DOWN.

## 4.2 3D mode

The output files generated in 3D mode runs have all the same format: the Gaussian Cube format. The grid points and atomic coordinates are given in the reference frame specified by the input (not the one given by the SIESTA calculation). The reference frame is orthogonal. These files can be used to draw 3D maps using programs like MOLDEN or MOLEKEL. Depending on the physical quantity being plotted, different files are generated:

### 4.2.1 Charge Density

If the calculation is not spin-polarized, two output files will be generated by DENCHAR:

*SystemLabel.RHO.cube* : Self-Consistent Charge Density at the points of the 3D grid.

*SystemLabel.DRHO.cube* : Difference between self-consistent charge density and the superposition of atomic densities.

If the calculation is spin-polarized, three output files will be generated by DENCHAR:

*SystemLabel.RHO.UP.cube* : Self-Consistent Charge Density of electrons with spin UP, at the points of the 3D grid.

*SystemLabel.RHO.DOWN.cube* : Self-Consistent Charge Density of electrons with spin DOWN, at the points of the 3D grid.

*SystemLabel.DRHO.cube* : Difference between self-consistent charge density and the superposition of atomic densities.

#### 4.2.2 Wavefunctions

DENCHAR plots each of the wavefunctions that is present in the file *SystemLabel.WFSX* generated by Siesta. For each wavefunction, a different file will be created containing the values of that wavefunction. In what follows, the wave function number is denoted by #.

If the calculation is not spin-polarized, one output file will be generated by DENCHAR for each wavefunction:

*SystemLabel.WF#.cube* : Values of the wave function number # in the 3D grid.

If the calculation is spin-polarized, there are two output files, namely:

*SystemLabel.WF#.UP.cube* : Values of the wave function number # with spin UP.

*SystemLabel.WF#.DOWN.cube* : Values of the wave function number # with spin DOWN.

## 5 EXAMPLES

In directory `~/siesta/Util/Denchar/Examples` you will find some examples of input files for the different options. The physical system is a cell of Si in the diamond structure at the experimental lattice constant. Also in `~/siesta/Util/Denchar/Tests` you will find some test cases with both input and output so that you can check that your installation reproduces the correct results. (This directory is not distributed with the program. You can get it from [http://fisica.ehu.es/ag/siesta-extra/denchar\\_tests.tgz](http://fisica.ehu.es/ag/siesta-extra/denchar_tests.tgz))