User's Guide

MACROAVE 1.1.0

June 07, 2003

Javier Junquera

Département de Physique, Université de Liège, Bâtiment B-5, B-4000 Sart-Tilman, Belgium

Pablo Ordejón

Institut de Ciència de Materials de Barcelona - CSIC, Campus de la U.A.B., 08193 Bellaterra, Barcelona, Spain

javier.junquera@ulg.ac.be

Contents

1 Introduction

The MACROAVE program implements the *macroscopic average technique*, introduced by A. Baldereschi and coworkers (A. Baldereschi, S. Baroni, and R. Resta, Phys. Rev. Lett. **61**, 734 (1988)). This is an extremely powerful method that relates microscopic quantities, typical outputs of first-principles codes, with macroscopic magnitudes, needed to perform electrostatic analysis. Within this methodology, we will be able of washing out all the wiggles of the rapidly-varying functions of position (resembling the underlying atomic structure) of the microscopic quantities, blowing up only the macroscopic features.

It is a basic tool to calculate some important magnitudes in surface or interface-related problems, such as:

- Band offsets and Work functions:
  L. Colombo, R. Resta and S. Baroni, Phys Rev B **44**, 5572 (1991)
- Effective charges:
  R. Martin and K. Kunc, Phys Rev B **24**, 2081 (1981)
- High-frequency dielectric constants:
  F. Bernardini and V. Fiorentini, Phys. Rev. B **58**, 15292 (1998)

MACROAVE reads the magnitude, $f(\vec{r})$, whose macroscopic average will be calculated (typically, charge densities or potentials) at the points of a three-dimensional uniform real space grid, as it is dumped into output files by standard first-principle codes. Then it performs the macroscopic average in a two step process:

- First: a planar average of $f(\vec{r})$ on planes parallel to the interface.
  (To establish the notation, we will call the plane parallel to the surface or the interface the (x, y) plane, whereas the perpendicular direction will be referred to as the z axis).

$$\overline{f}(z) = \frac{1}{S}\int_S f(\vec{r})\, dxdy \tag{1}$$

  where S is the surface of the unit cell perpendicular to the given direction.

- Second: a final convolution of $\overline{f}(z)$ with filter functions. We choose step functions, Θ, of length l.

$$\omega_l(z) = \frac{1}{l}\Theta\left(\frac{l}{2} - |z|\right) \tag{2}$$

$$\overline{\overline{f}}(z) = \int dz' \int dz'' \omega_{l_1}(z - z')\, \omega_{l_2}(z' - z'')\, \overline{f}(z'') \tag{3}$$

Currently, MACROAVE can handle directly the microscopic information provided by SIESTA and ABINIT, but it should be easily adapted to any other first-principle code.

Coded by J. Junquera and P. Ordejón, April 1999

Adapted for ABINIT by J. Junquera, October 2002

This is a short description of the compilation procedures and of the datafile format for the MACROAVE code. This version is a very preliminary release of the code. Please report problems, bugs and suggestions to javier.junquera@ulg.ac.be

2 Compilation

In this section we give all the steps required to install the program. We assume that you use UNIX and you will install MACROAVE in ∼/Macroave/Src, where ∼ indicates your home directory. The commands you must type are in `typewriter` font, and `$` indicates the prompt.

Uncompress and unpack the gzipped tar file macroave.tar.gz

```
$ gunzip macroave.tar.gz
```

```
$ tar -xvf macroave.tar
```

Now, we suppose you have a fortran-90 compiler and you want to compile the source files, included in the directory `Macroave/Src`. Go into this subdirectory

```
$ cd Macroave/Src
```

The compilation of the program is done using a Makefile that is provided with the code. This Makefile will generate the executable file in several architectures, with a minimum of tuning required from the user. The only variables you must set up (included in a file called arch.make) are:

- FC : location of the fortran-90 compiler.
- FFLAGS : flags to optimize the compilation in your platform.

Edit the arch.make file and set up the options that suit your platform.

Then, you just need to run the 'make' utility as usual, typing make in the source directory:

```
$ make
```

Since MACROAVE is written in fortran-90 and the memory is allocated dynamically, the executable file, called macroave, should work for any job.

3 Running the program

As it was mentionned in the introduction (Section 1), MACROAVE needs as input the microscopic magnitude, $f(\vec{r})$, whose macroscopic average we want to calculate. $f(\vec{r})$ will be, typically, a charge density or a given potential (electrostatic, exchange-correlation only, total,...). This information, that will be supplied by a first-principles electronic-simulation code, is usually stored at the points of a three-dimensional real-space grid.

Obviously, the first thing we must do is to run the electronic-simulation program for the system we are interested in, setting up the variables that instruct to write the corresponding magnitude. At the current time, MACROAVE is able to digest directly the output files supplied by SIESTA and ABINIT. The relevant input variables *in these first-principles codes* are:

- SIESTA
  - SaveRho
  - SaveDeltaRho
  - SaveElectrostaticPotential
  - SaveTotalPotential
  - SaveIonicCharge
  - SaveTotalCharge
  - LocalDensityOfStates
- ABINIT
  - prtpot
  - prtvha
  - prtvhxc
  - prtvxc
  - prtden

We refer the reader to the User's Guide of SIESTA or ABINIT to learn more about these different options.

Once the simulation is finished, and the relevant output files written, then move to the directory where the job was run (let's call it ∼`/rundir`)

```
cd ∼/rundir
```

Edit the MACROAVE's input file (called *macroave.in*) and set up the right values for the different variables. This file will be fully explained in section 4.

Execute macroave.x

```
$ ∼/Macroave/Src/macroave.x
```

The output is dumped in files which will be described in Section 5.

4 Input data file

Apart from the information taken from the electronic-simulation code, MACROAVE requires only an input data file, named *macroave.in*.

This input file has eigth lines:

first line (*string*): Name of the first-principles code used to generate the microscopic magnitude, $f(\vec{r})$. At present, it only accepts two options:

- Siesta
- Abinit

second line (*string*): Microscopic magnitude whose macroscopic average will be calculated:

- Potential
- Charge

third line (*string*): Name of the file (output of the first-principles code) where the magnitude $f(\vec{r})$ is stored. In the case of SIESTA, only the **SystemLabel** is required (see SIESTA User's Guide).

fourth line (*integer*): Number of convolutions with step functions required to perform the macroscopic average. It can take only two different values:

- 1 (for surface-related problems).
- 2 (for interface-related problems).

fifth line (*real*): Length of the first step function used to perform the macroscopic average (see Eq. 2)

Units: bohrs

sixth line (*real*): Length of the second step function used to perform the macroscopic average (see Eq. 2)

Units: bohrs

Use: Only use if the number of convolutions is equal to 2.

seventh line (*integer*): Electronic charge of the system

Units: electrons

Use: Only use if we are computing the macroscopic average of charge densities.

eigth line (*string*): Kind of interpolation to get $f(\vec{r})$ at a fine FFT grid, starting from the grid used in the first-principles code.

At the current time, it only accepts two different values

- Spline
- Linear

5 Output files

Two output files are produced, containing the information about the planar (see Eq. 1) and the macroscopic average (see Eq. 3) of $f(\vec{r})$.

Contains, in two colums, values of z and the profile of the planar or macroscopic average.

The name of these output files is the same as the one introduced in the third line of the input, plus an extension:

.PAV for the planar average.

Units:

- electrons/bohr^3 if $f(\vec{r})$ is a charge density.
- eV if $f(\vec{r})$ is a potential. density.

.MAV for the macroscopic average.

Units:

- electrons/bohr^3 if $f(\vec{r})$ is a charge density.
- eV if $f(\vec{r})$ is a potential. density.

6 Examples

In directory `~/Macroave/Examples` you will find some examples of input files.

7 Known bugs and errors

- The code only works for orthorhombic unit cells.
- Spin polarization not implemented yet. The planar average, and the corresponding macroscopic average are only implemented for the first component of array RHO.