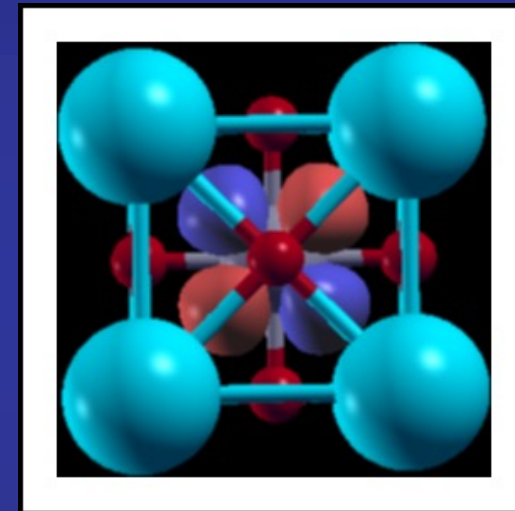
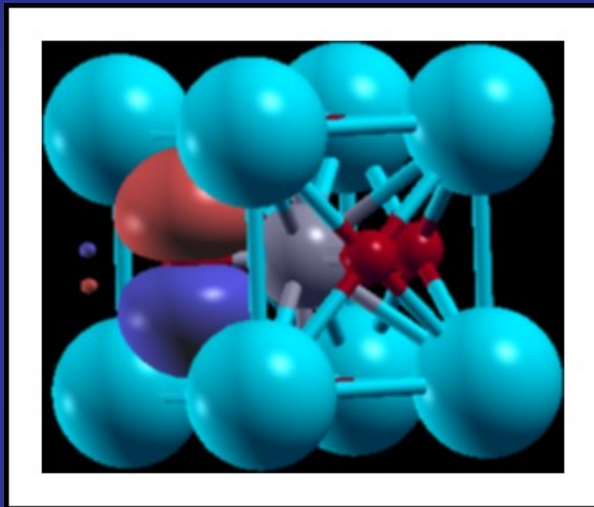


# How to prepare the interface between SIESTA and WANNIER90



**Javier Junquera**

# Important bibliography:

For a review on Maximally Localized Wannier functions:

REVIEWS OF MODERN PHYSICS, VOLUME 84, OCTOBER–DECEMBER 2012

## **Maximally localized Wannier functions: Theory and applications**

Nicola Marzari

*Theory and Simulation of Materials (THEOS), École Polytechnique Fédérale de Lausanne,  
Station 12, 1015 Lausanne, Switzerland*

Arash A. Mostofi

*Departments of Materials and Physics, and the Thomas Young Centre for Theory  
and Simulation of Materials, Imperial College London, London SW7 2AZ, United Kingdom*

Jonathan R. Yates

*Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH,  
United Kingdom*

Ivo Souza

*Centro de Física de Materiales (CSIC) and DIPIC, Universidad del País Vasco,  
20018 San Sebastián, Spain and Ikerbasque Foundation, 48011 Bilbao, Spain*

David Vanderbilt

*Department of Physics and Astronomy, Rutgers University, Piscataway,  
New Jersey 08854-8019, USA*

# Last features implemented in the WANNIER90 code

<http://www.wannier.org>

J. Phys.: Condens. Matter **32** (2020) 165902 (25pp)

<https://doi.org/10.1088/1361-648X/ab51ff>

## Wannier90 as a community code: new features and applications

Giovanni Pizzi<sup>1,29,30</sup>, Valerio Vitale<sup>2,3,29</sup>, Ryotaro Arita<sup>4,5</sup>,  
Stefan Blügel<sup>6</sup>, Frank Freimuth<sup>6</sup>, Guillaume Géranton<sup>6</sup>,  
Marco Gibertini<sup>1,7</sup>, Dominik Gresch<sup>8</sup>, Charles Johnson<sup>9</sup>,  
Takashi Koretsune<sup>10,11</sup>, Julen Ibañez-Azpiroz<sup>12</sup>, Hyungjun Lee<sup>13,14</sup>,  
Jae-Mo Lihm<sup>15</sup>, Daniel Marchand<sup>16</sup>, Antimo Marrazzo<sup>1</sup>,  
Yuriy Mokrousov<sup>6,17</sup>, Jamal I Mustafa<sup>18</sup>, Yoshiro Nohara<sup>19</sup>,  
Yusuke Nomura<sup>4</sup>, Lorenzo Paulatto<sup>20</sup>, Samuel Poncé<sup>21</sup>,  
Thomas Ponweiser<sup>22</sup>, Junfeng Qiao<sup>23</sup>, Florian Thöle<sup>24</sup>,  
Stepan S Tsirkin<sup>12,25</sup>, Małgorzata Wierzbowska<sup>26</sup>, Nicola Marzari<sup>1,29</sup>,  
David Vanderbilt<sup>27,29</sup>, Ivo Souza<sup>12,28,29</sup>, Arash A Mostofi<sup>3,29</sup>  
and Jonathan R Yates<sup>21,29</sup>

# Important bibliography:

The user guide and tutorial of the WANNIER90 code

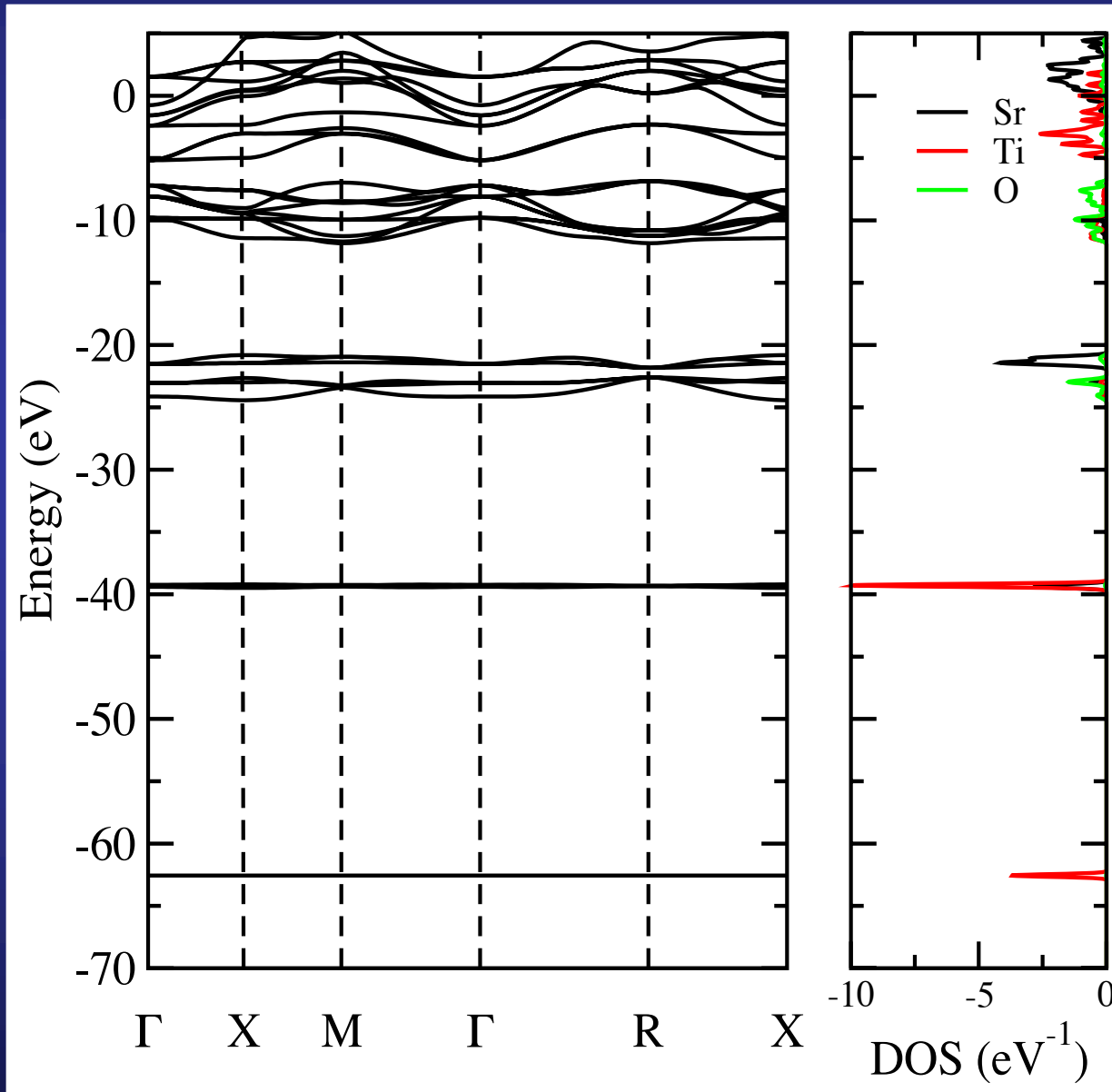
wannier90: Tutorial

Version 3.1

Freely available from:  
<http://www.wannier.org>

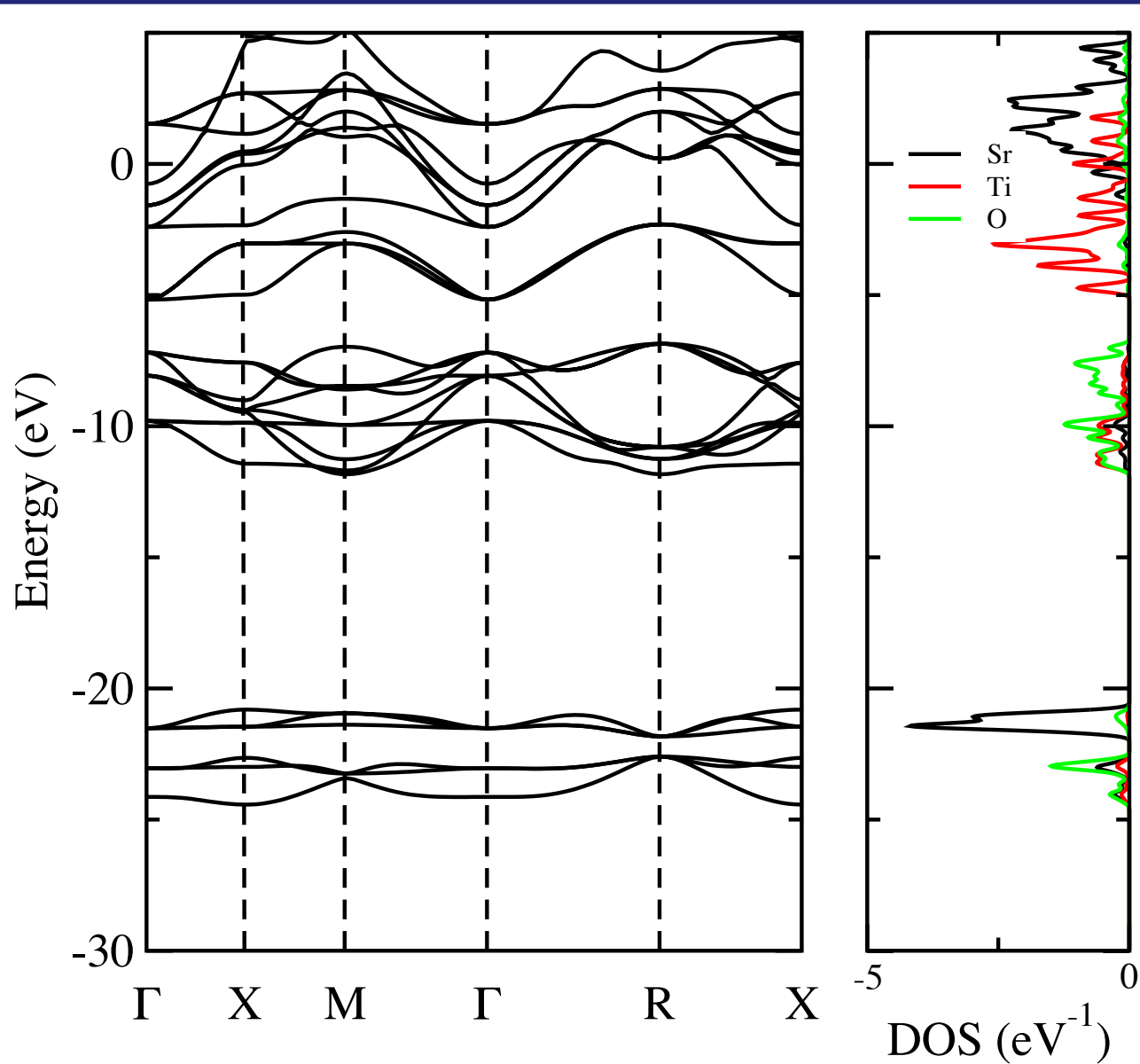
# After running SIESTA and compute the PDOS, we can analyze the character of the different bands

Which atoms contribute more to the bands at a particular energy window



# We can analyze the character of the different bands

Which atoms contribute more to the bands at a particular energy window  
Zoom around the top of the valence bands and bottom of conduction bands



**Bottom of conduction bands: mostly Ti character**

**Top of valence bands: mostly O character**

**We can project on particular atomic orbitals within an atom to further define the character.**

# Flowchart to compute Maximally Localized Wannier functions from the input provided by SIESTA

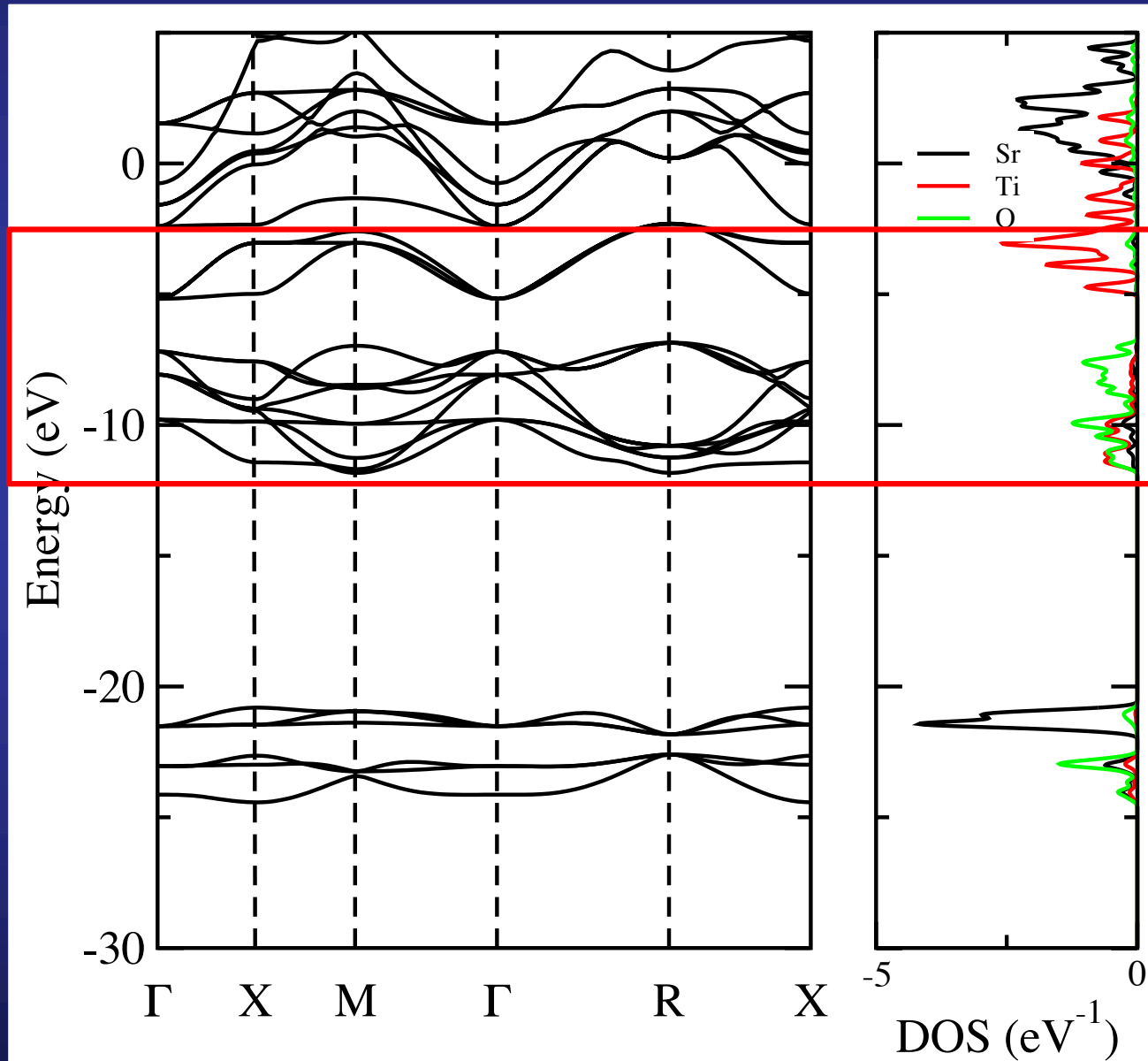
1. Prepare compatible input files for SIESTA and WANNIER90
2. Run WANNIER90 in pre-processing format
3. Run SIESTA and generate the required files to feed WANNIER90
4. Change the extension to the SystemLabel.eigW file
5. Run WANNIER90 to minimize the localization functional, plot the Wanniers, interpolate bands, etc as a post-processing tool

# Flowchart to compute Maximally Localized Wannier functions from the input provided by SIESTA

1. Prepare compatible input files for SIESTA and WANNIER90
2. Run WANNIER90 in pre-processing format
3. Run SIESTA and generate the required files to feed WANNIER90
4. Change the extension to the SystemLabel.eigW file
5. Run WANNIER90 to minimize the localization functional, plot the Wanniers, interpolate bands, etc as a post-processing tool



# Choose the Bloch states that will be used to compute the Wannier functions



In this particular example, we are interested in the wannierization of:

**Bottom of conduction bands: mostly Ti character**

**Top of valence bands: mostly O character**

That means:

- 3 O  $p$  bands ( $p_x, p_y, p_z$ )  
× 3 O atoms
- 3 Ti  $t_{2g}$  bands ( $d_{xy}, d_{yz}, d_{xz}$ )

**12 bands to wannierize**

# Prepare compatible input files for SIESTA and WANNIER90: How to select the number of bands to wannierized

## seedname.win file (input of WANNIER90)

```
num_bands = 12 ! Number of bands in the first-principles calculation
! used to form the overlap matrices
! seedname.mmn file
! Default: num_bands = num_wann
num_wann = 12 ! Number of Wannier functions to be found
! There must be as many projectors in the
! projections block as num_wann functions here.
! If num_wann < num_bands, then the disentanglement
! procedure is activated
! In this example, we want to Wannierize the
! top of the valence bands (0 2p in character)
! and the bottom of the conduction bands (Ti t2g),
! so it amounts to
! - 0 p bands (3 bands * 3 0 atoms)
! - Ti t2g bands (3 bands: dxy, dyz, dxz)
! Twelve bands in total
!
! For every k-point, the bands in Siesta are ordered from lowest to highest
! energy
! The character of the bands are known after computation of the PDOS.
! In order, for the case of SrTiO3:
! -----
! Bands to be excluded
! -----
! Band 1 (lowest band): Ti-3s character
! Bands 2, 3, and 4: Ti-3p character (px, py, and pz)
! Band 5: Sr-4s character
! Bands 6, 7, and 8: 0-2s character (1 band * 3 0 atoms in the unit cell)
! Band 9, 10, and 11: Sr-4p character (px, py, and pz)
! -----
! Bands to be wannierized
! -----
! Bands 12-20: 0-2p character (3 bands * 3 0 atoms in the unit cell)
! Bands 21, 22, and 23: Ti-t2g characer (dxy, dyz, dxz)
!
! In exclude_bands, we list the bands to excluded from the calculation
! Here we exclude all bands except the 02p and Ti t2g bands.
! That means that we exclude:
! - Ti 3s band (1 band)
! - Ti 3p bands (3 bands)
! - Sr 4s bands (1 band)
! - Sr 4p bands (3 bands)
! - 0 2s bands (1 band * 3 atoms = 3 bands)
! This amounts to the first eleven bands for every k-point.
exclude_bands : 1,2,3,4,5,6,7,8,9,10,11
```

## seedname.fdf file (input of SIESTA)

```
Siesta2Wannier90.WriteMmn .true.
Siesta2Wannier90.WriteAmn .true.
Siesta2Wannier90.WriteEig .true.
Siesta2Wannier90.WriteUnk .true.
Siesta2Wannier90.NumberOfBands 23
Siesta2Wannier90.UnkGrid1 20
Siesta2Wannier90.UnkGrid2 20
Siesta2Wannier90.UnkGrid3 20
```

Select the same number of bands here as required to include all the bands.

Internally, SIESTA discards those listed in `exclude_bands` and produce the overlap matrices only for the bands to be wannierized

# Prepare compatible input files for SIESTA and WANNIER90: The atomic geometry

The unit cells lattice vectors and atomic positions must be the same in the WANNIER90 input and in the SIESTA input used to generate the projections

**seedname.win** file  
(input of WANNIER90)

```
begin unit_cell_cart
Ang
3.874 0.000 0.000
0.000 3.874 0.000
0.000 0.000 3.874
end unit_cell_cart

begin atoms_frac
Sr 0.0 0.0 0.0
Ti 0.5 0.5 0.5
O 0.5 0.5 0.0
O 0.5 0.0 0.5
O 0.0 0.5 0.5
end atoms_frac
```

**seedname.fdf** file  
(input of SIESTA)

```
LatticeConstant 3.874 Ang
%block LatticeVectors
1.000 0.000 0.000
0.000 1.000 0.000
0.000 0.000 1.000
%endblock LatticeVectors

AtomicCoordinatesFormat Fractional
%block AtomicCoordinatesAndAtomicSpecies
0.00000000 0.00000000 0.00000000 1 87.62 Sr
0.50000000 0.50000000 0.50000000 2 47.867 Ti
0.50000000 0.50000000 0.00000000 3 15.9994 O
0.50000000 0.00000000 0.50000000 3 15.9994 O
0.00000000 0.50000000 0.50000000 3 15.9994 O
%endblock AtomicCoordinatesAndAtomicSpecies
```

The unit cell lattice vectors must be the same in  
symmetry and length of the cell vectors

The coordinates of the atoms must be the same,  
as well as the order in which they are entered

# Prepare compatible input files for SIESTA and WANNIER90: The WANNIER90 input file: k-points block

SIESTA must compute the overlap of the periodic part of the wave functions at neighbour k-points

$$M_{mn}^{(\vec{k}, \vec{b})} = \langle u_{m\vec{k}} | u_{n\vec{k}+\vec{b}} \rangle$$

**seedname.win** file  
(input of WANNIER90)  
to indicate which  $\vec{k}$  points  
will be considered

$\vec{b}$  is the vector which connect a  
given  $\vec{k}$  with its neighbours

```
mp_grid : 4 4 4

begin kpoints
 0.00000000  0.00000000  0.00000000
 0.00000000  0.00000000  0.25000000
 0.00000000  0.00000000  0.50000000
 0.00000000  0.00000000  0.75000000
 0.00000000  0.25000000  0.00000000
 0.00000000  0.25000000  0.25000000
 0.00000000  0.25000000  0.50000000
 0.00000000  0.25000000  0.75000000
 0.00000000  0.50000000  0.00000000
 0.00000000  0.50000000  0.25000000
 0.00000000  0.50000000  0.50000000
 0.00000000  0.50000000  0.75000000
 0.00000000  0.75000000  0.00000000
 0.00000000  0.75000000  0.25000000
 0.00000000  0.75000000  0.50000000
 0.00000000  0.75000000  0.75000000
  ...
 0.75000000  0.75000000  0.75000000
end kpoints
```

Dimensions of the regular Monkhorst-Pack  $\vec{k}$ -point grid mesh

Each line gives the coordinate of a  $\vec{k}$ -point in relative units, i.e. in fractional units with respect to the primitive reciprocal lattice vector

This list of  $\vec{k}$ -points is independent of the  
Monkhorst-Pack mesh used to achieve the self-consistency

# Prepare compatible input files for SIESTA and WANNIER90: The WANNIER90 input file: k-points block

SIESTA must compute the overlap of the periodic part of the wave functions at neighbour k-points

$$M_{mn}^{(\vec{k}, \vec{b})} = \langle u_{m\vec{k}} | u_{n\vec{k}+\vec{b}} \rangle$$

**seedname.win** file  
(input of WANNIER90)  
to indicate which  $\vec{k}$  points  
will be considered

$\vec{b}$  is the vector which connect a  
given  $\vec{k}$  with its neighbours

**seedname.fdf** file  
(input of SIESTA)

```
mp_grid : 4 4 4

begin kpoints
 0.00000000 0.00000000 0.00000000
 0.00000000 0.00000000 0.25000000
 0.00000000 0.00000000 0.50000000
 0.00000000 0.00000000 0.75000000
 0.00000000 0.25000000 0.00000000
 0.00000000 0.25000000 0.25000000
 0.00000000 0.25000000 0.50000000
 0.00000000 0.25000000 0.75000000
 0.00000000 0.50000000 0.00000000
 0.00000000 0.50000000 0.25000000
 0.00000000 0.50000000 0.50000000
 0.00000000 0.50000000 0.75000000
 0.00000000 0.75000000 0.00000000
 0.00000000 0.75000000 0.25000000
 0.00000000 0.75000000 0.50000000
 0.00000000 0.75000000 0.75000000
  ...
 0.75000000 0.75000000 0.75000000
end kpoints
```

```
Siesta2Wannier90.WriteMmn .true.
Siesta2Wannier90.WriteAmn .true.
Siesta2Wannier90.WriteEig .true.
Siesta2Wannier90.WriteUnk .true.

Siesta2Wannier90.NumberOfBands 23

Siesta2Wannier90.UnkGrid1 20
Siesta2Wannier90.UnkGrid2 20
Siesta2Wannier90.UnkGrid3 20
```

The information is passed through the **seedname.nnkp** file,  
generated by WANNIER90 after running in pre-processing mode

# Prepare compatible input files for SIESTA and WANNIER90: The WANNIER90 input file: projections

SIESTA must compute the overlap of the Bloch states  $|\psi_{m\vec{k}}\rangle$  onto trial localised orbitals  $|g_n\rangle$

$$A_{mn}^{(\vec{k})} = \langle \psi_{m\vec{k}} | g_n \rangle$$

Each projection is associated with a site and an angular momentum state defining the projection function

**seedname.win** file  
(input of WANNIER90)  
to indicate the projection  
functions

In our example, we are going to project over:  
the  $p$ -states of all the O and  
the  $t_{2g}$  orbitals of Ti

```
guiding_centres = T      ! The projection centres are used as the guiding
                        !   centres in the Wannierisation routines

begin projections       ! The projections block defines a set of
O:p                    !   localised functions used to generate an
Ti:l=2,mr=2,3,5       !   initial guess for the unitary transformations.
end projections
```

Take a look to the WANNIER90 User's Guide (pages 33-38)  
to check the syntax and short-cuts

# Prepare compatible input files for SIESTA and WANNIER90: The WANNIER90 input file: projections

SIESTA must compute the overlap of the Bloch states  $|\psi_{m\vec{k}}\rangle$  onto trial localised orbitals  $|g_n\rangle$

$$A_{mn}^{(\vec{k})} = \langle \psi_{m\vec{k}} | g_n \rangle$$

Each projection is associated with a site and an angular momentum state defining the projection function

**seedname.win** file  
(input of WANNIER90)  
to indicate the projection  
functions

```
guiding_centres = T

begin projections
0:p
Ti:l=2,mr=2,3,5
end projections
```

**seedname.fdf** file  
(input of SIESTA)

```
Siesta2Wannier90.WriteMmn .true.
Siesta2Wannier90.WriteAmn .true.
Siesta2Wannier90.WriteEig .true.
Siesta2Wannier90.WriteUnk .true.

Siesta2Wannier90.NumberOfBands 23

Siesta2Wannier90.UnkGrid1 20
Siesta2Wannier90.UnkGrid2 20
Siesta2Wannier90.UnkGrid3 20
```

The information is passed through the **seedname.nnkp** file,  
generated by WANNIER90 after running in pre-processing mode

# Flowchart to compute the Maximally Localized Wannier functions from the input provided by siesta

1. Prepare compatible input files for SIESTA and WANNIER90

2. Run WANNIER90 in pre-processing format

```
$ wannier90.x -pp SrTiO3
```

This generates a seedname.nnkp file that is digested by SIESTA



# Flowchart to compute the Maximally Localized Wannier functions from the input provided by siesta

1. Prepare compatible input files for SIESTA and WANNIER90

2. Run WANNIER90 in pre-processing format

```
$ wannier90.x -pp SrTiO3
```

3. Run SIESTA and generate the required files to feed WANNIER90

```
$ siesta < SrTiO3.fdf > SrTiO3.out
```

# Successful output of SIESTA

```
siesta2wannier90: Reading the SrTiO3.nnkp file

read_nnkp: Checking info from the SrTiO3.nnkp file
read_nnkp: Reading data about real lattice
read_nnkp:      - Real lattice is ok
read_nnkp: Reading data about reciprocal lattice
read_nnkp:      - Reciprocal lattice is ok
read_nnkp: Reading data about k-points
read_nnkp: Reading data about projection centers
read_nnkp: Reading data about k-point neighbours
read_nnkp: Reading data about excluded bands

noccbands: Total number of electrons      40
noccbands: Number of occupied bands      20
number_bands_wannier: Number of bands for wannierization
number_bands_wannier: before excluding bands      =      23      0

Number of bands for wannierization after excluding bands:      12      0
Bands to be wannierized:
    12   13   14   15   16   17   18   19   20   21   22   23

compute_pw_matrix: Computing the matrix elements of a plane wave

mmn: Overlap matrices between periodic part of wavefunctions
mmn: written in SrTiO3.mmn file

amn: Overlap matrices between trial projection functions and wavefunctions
amn: written in SrTiO3.amn file

eig: Eigenvalues of the Hamiltonian
eig: written in SrTiO3.eigW file

siesta2wannier90: All the information dumped in the corresponding files
siesta2wannier90: End of the interface between Siesta and Wannier90
```

# Flowchart to compute the Maximally Localized Wannier functions from the input provided by siesta

1. Prepare compatible input files for SIESTA and WANNIER90

2. Run WANNIER90 in pre-processing format

```
$ wannier90.x -pp SrTiO3
```

3. Run SIESTA and generate the required files to feed WANNIER90

```
$ siesta < SrTiO3.fdf > SrTiO3.out
```

4. Change the extension to the SystemLabel.eigW file

```
$ mv SrTiO3.eigW SrTiO3.eig  
overwrite SrTiO3.eig? (y/n [n]) y
```

# Flowchart to compute the Maximally Localized Wannier functions from the input provided by siesta

1. Prepare compatible input files for SIESTA and WANNIER90

2. Run WANNIER90 in pre-processing format

```
$ wannier90.x -pp SrTiO3
```

3. Run SIESTA and generate the required files to feed WANNIER90

```
$ siesta < SrTiO3.fdf > SrTiO3.out
```

4. Change the extension to the SystemLabel.eigW file

```
$ mv SrTiO3.eigW SrTiO3.eig  
overwrite SrTiO3.eig? (y/n [n]) y
```

5. Run WANNIER90 to minimize the localization functional, plot the Wanniers, interpolate bands, etc as a post-processing tool

```
$ wannier90.x SrTiO3
```

# Successful output of WANNIER90

Cycle: 100

WF centre and spread	1	( 1.937000, 1.937000, -0.000000 )	1.14103775
WF centre and spread	2	( 1.937000, 1.937000, -0.000002 )	1.01254313
WF centre and spread	3	( 1.937000, 1.937000, 0.000000 )	1.01253811
WF centre and spread	4	( 1.936999, 0.000001, 1.937000 )	1.01254138
WF centre and spread	5	( 1.937001, 0.000000, 1.937000 )	1.01254288
WF centre and spread	6	( 1.937001, -0.000001, 1.937001 )	1.14104145
WF centre and spread	7	( 0.000001, 1.937000, 1.937000 )	1.01254154
WF centre and spread	8	( -0.000001, 1.937001, 1.937000 )	1.14104851
WF centre and spread	9	( -0.000001, 1.937001, 1.937000 )	1.01253866

9 WF centered  
on the three O

WF centre and spread	10	( 1.937000, 1.937001, 1.937000 )	0.90476826
WF centre and spread	11	( 1.937000, 1.937000, 1.936999 )	0.90476759
WF centre and spread	12	( 1.937000, 1.937000, 1.937000 )	0.90477667

3 WF centered  
on Ti

Sum of centres and spreads ( 17.433000, 17.433003, 17.432997 ) 12.21268593

100	-0.178E-14	0.0000001797	12.2126859319	1.04	<-- CONV
-----	------------	--------------	---------------	------	----------

Check that the minimization  
has converged

# How to plot the Wannier functions

First of all, SIESTA has to write the periodic part of the Bloch functions in a 3D grid. The number of points in the grid along the three lattice vectors are given by

**seedname.fdf** file  
(input of SIESTA)

```
Siesta2Wannier90.UnkGrid1      20
Siesta2Wannier90.UnkGrid2      20
Siesta2Wannier90.UnkGrid3      20
```

This produces many files with the name UNKXXXXX.Y where

- XXXXX is the number of the k-point, from 1 to the number of points included in seedname.win file
- Y refers to the spin component (1 or 2)

**seedname.win** file  
(input of WANNIER90)

```
!To plot the WF
wannier_plot = T                ! If true, then the code will write out the
                                ! Wannier functions in a super-cell whose size is
                                ! and in a format specified by wannier_plot_format

wannier_plot_supercell = 3
wannier_plot_list = 1,10       ! A list of WF to plot.
                                ! The WF numbered as per the seedname.wout
                                ! file after the minimisation of the spread.

restart = plot
```

```
$ wannier90.x SrTiO3
```

# How to plot the Wannier functions

WANNIER90 produces files with the name:  
SrTiO3\_00002.xsf, SrTiO3\_00010.xsf that can be directly plotted with XCRYSDEN

Once XCrySDen starts, click on  
**Tools → Data Grid**

Click on **OK**

Then, select:

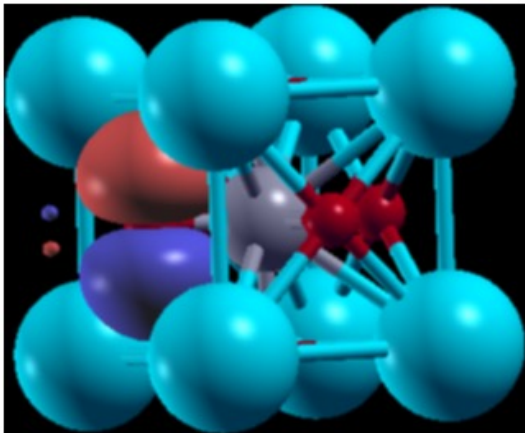
**Degree of triCubic Spline: 3**

**Click on Render+/- isovalue**

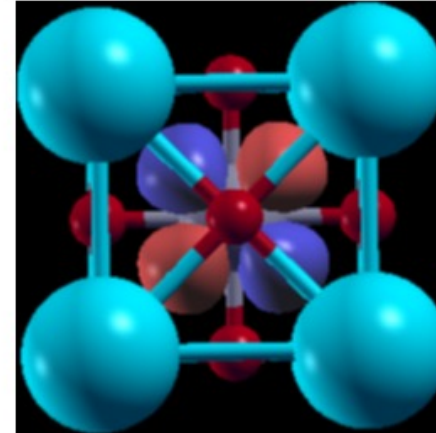
**Select the desired isovalue (in this example 0.1)**

**Submit**

**SrTiO3\_00002.xsf**



**SrTiO3\_00010.xsf**



# How to plot the interpolated band structure obtained from the subspace spanned by the Wannier function

**Seedname.win** file  
(input of WANNIER90)

```
begin kpoint_path                                ! Defines the path in k-space along
G 0.000  0.000 0.000 X 0.500  0.000 0.000      ! which to calculate the
X 0.500  0.000 0.000 M 0.500  0.500 0.000      ! bandstructure. Each line gives the
M 0.500  0.500 0.000 G 0.000  0.000 0.000      ! start and end point (with labels)
G 0.000  0.000 0.000 R 0.500  0.500 0.500      ! for a section of the path.
R 0.500  0.500 0.500 X 0.500  0.000 0.000      ! Values are in fractional
end kpoint_path                                  ! coordinates with respect to the
                                                  ! primitive reciprocal lattice
                                                  ! vectors.

bands_plot =T
```

```
$ wannier90.x SrTiO3
```

This will produce three files that can be directly plotted by gnuplot

SrTiO3\_band.gnu

SrTiO3\_band.kpt

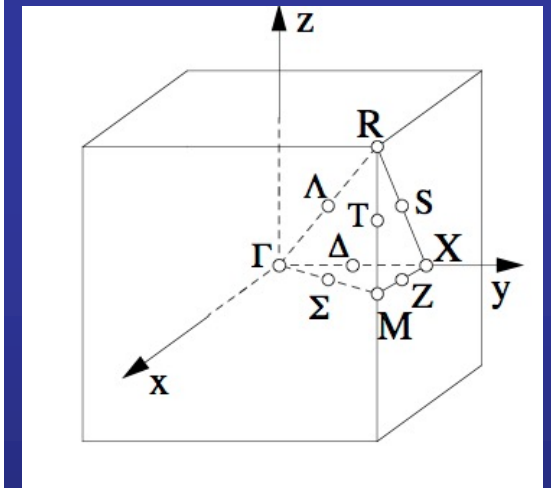
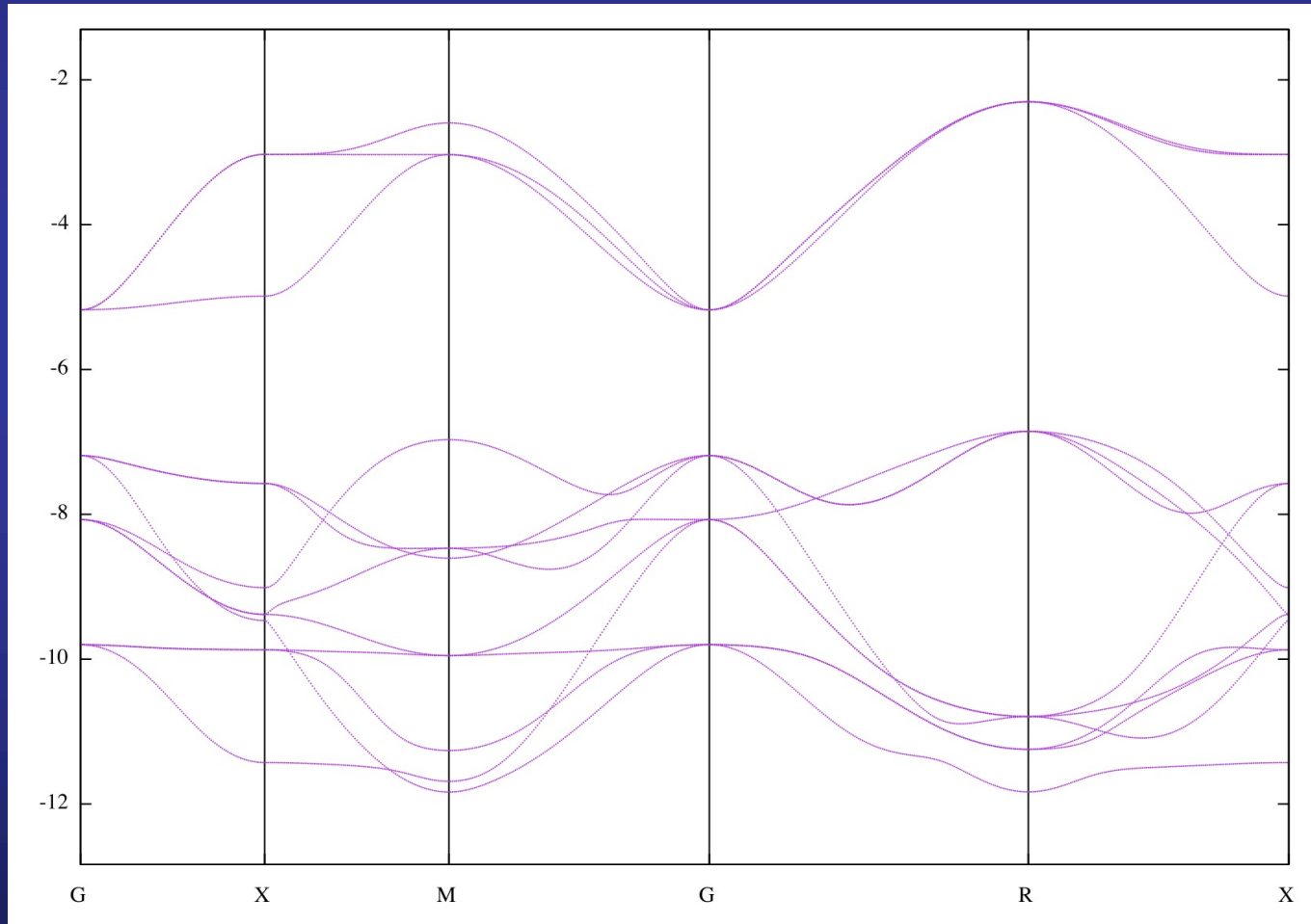
SrTiO3\_band.dat

(in the last version of gnuplot, it is very likely that you have to edit SrTiO3\_band.gnu and change the first line to set style data dots)



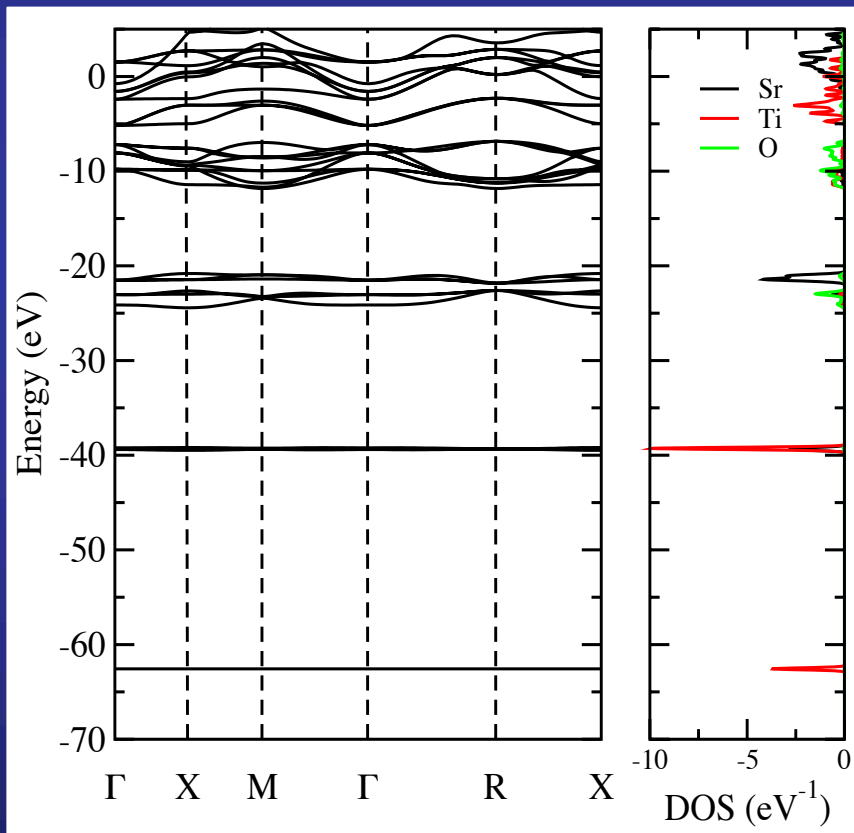
# How to plot the interpolated band structure obtained from the subspace spanned by the Wannier function

```
$ gnuplot  
$ load "SrTiO3_band.gnu"
```

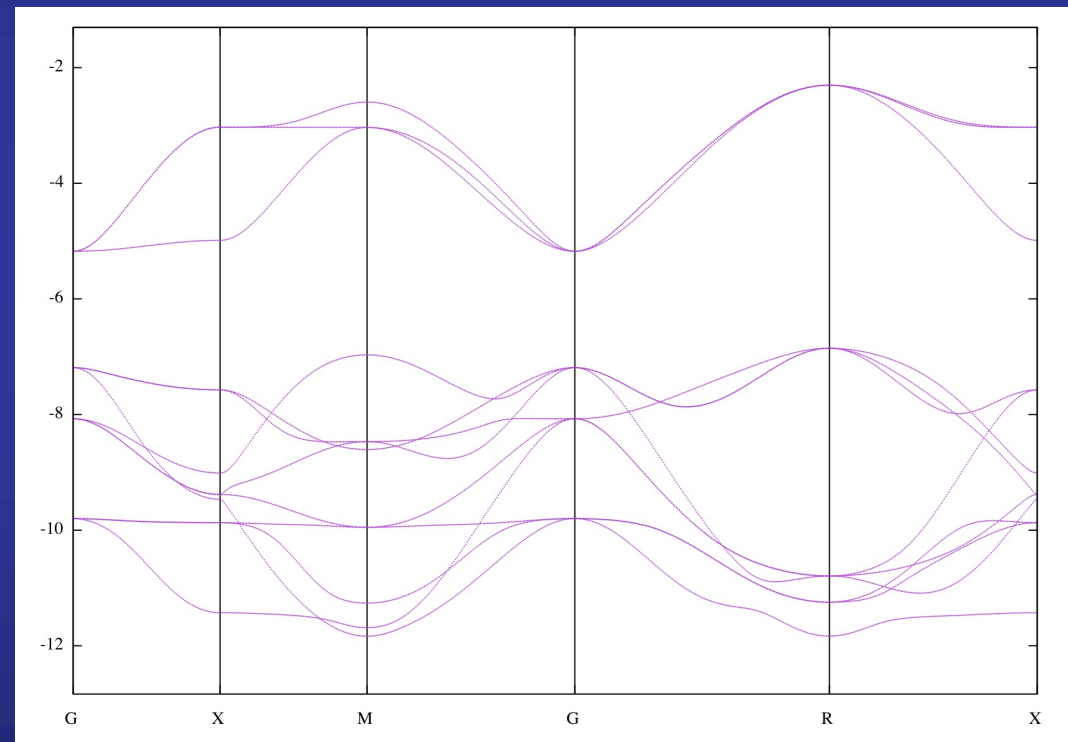


# How to plot the interpolated band structure obtained from the subspace spanned by the Wannier function

From SIESTA



Interpolated from WANNIER90



Only the top of the valence band and the bottom of the conduction band

# Funding

## SPANISH INITIATIVE FOR ELECTRONIC SIMULATIONS WITH THOUSANDS OF ATOMS: CÓDIGO ABIERTO CON GARANTÍA Y SOPORTE PROFESIONAL: SIESTA-PRO

Proyecto financiado por el Ministerio de Economía, Industria y Competitividad,  
y cofinanciado con Fondos Estructurales de la Unión Europea

Referencia: RTC-2016-5681-7

Objetivo Temático del Programa Operativo:

"Promover el desarrollo tecnológico, la innovación y una investigación de calidad"

