Practical Session:

Electron model building

Pablo García-Fernández

Santander July 2017

Universidad de Cantabria

Goals for the practical session

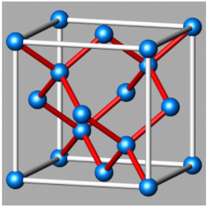
Overall objective: Generate models for Si

1. Introduction
2. Training set:
3. Running WANNIER90
   - Selecting the windows
   - sp^3 or s3p
   - Visualize orbitals
4. Running MODELMAKER
   - The folder structure
   - mode 1 - cutoff bands
   - mode 2 - occupations
   - mode 3 - generate terms
   - mode 4 - fit

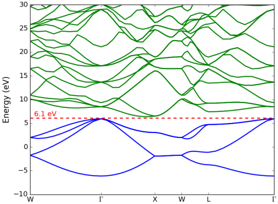
Introduction

We seek to build a model for silicon

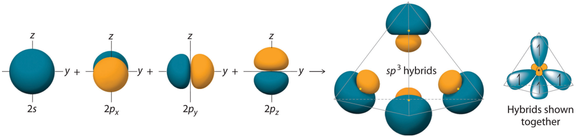
Structure



Electronic structure



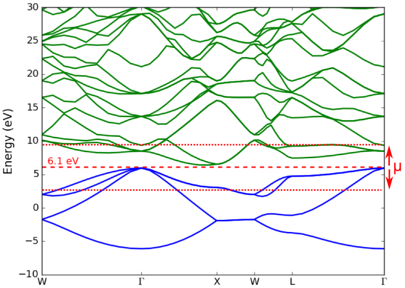
1. The structure is Fd-3m with 2 atoms in the primitive cell
2. It is a semiconductor with small indirect gap
3. Textbook valence + conduction bands → Si(3s)+Si(3p)



https://2012books.lardbucket.org/books/principles-of-general-chemistry-v1.0

sp^3 hybridization invoked along tetrahedral coordination

Silicon: Training set



- We will be using LDA
  Very small gap!
  Delocalized electrons
  no magnetism
- Fixed experimental geometry
- Dope electrons and holes
  holes→ explore valence
  electrons→ conduction

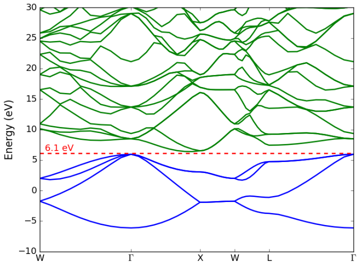
Going through full procedure, FP and SP, very time consuming:

1. First step Generate reasonable WF with WANNIER90
2. Second step Use pre-generated data to build model
   No magnetism ⟶ No J
   Fixed geometry ⟶ No electron-phonon
3. If time allows use the data you generated

The WANNIER90 input

The bands for silicon using VASP are:

```
num_wann =   8   ! 2at*(Si(3s)+3 Si(3p))
num_bands =  32
dis_win_min  =  ????
dis_win_max  =  ????
dis_froz_min =  ????
dis_froz_max =  ????
Begin Projections
Si:????
End Projections
guiding_centres = T
hr_plot = T
wannier_plot = T
```



LDA silicon bands

- num_wann is the number of Wanniers we want to get
- num_bands is the number of bands in the calculation (12)
- num_iter is the number of iterations in the Wannier optimization
- wannier_plot activates exporting a file to plot the Wanniers using VESTA
- dis_win_max//min the energy window where we project the bands
- dis_froz_max//min the energy window where we the full band is put into WFs
- Projections is a block where we feed an initial guess of the Wanniers
  The format is:
  Center: Atoms (Si) or position in fractional f=x,y,z
  Type of orbital: s, px, py, pz, sp, sp3, sp2, ...

WANNIER90: Energy windows and projections

What kind of model should we construct...?

sp^3 model?

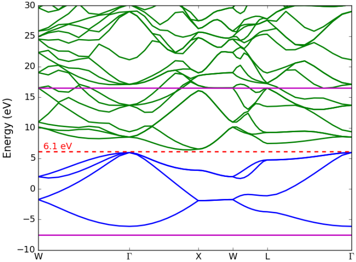
```
Begin Projections
Si:sp3
End Projections
```

s3p model?

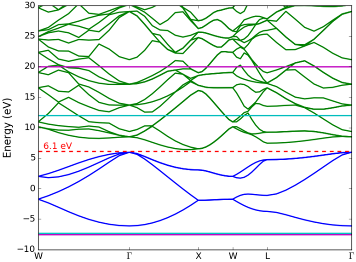
```
Begin Projections
Si:s;px;py;pz
End Projections
```

Where should we place the energy windows?

Small window?



Large window and freeze?



Wannier90: Running examples

Connect to the cluster:

```
$ ssh -X citimacguest@tejo.unican.es
```

Password: $spDft17&

```
$ cd workshop/myname
$cp input/* .
$vi wannier90.win
```

Modify the windows and projections and launch the calculation

```
$ ./launch_vasp_wannier90
```

Wait and inspect the wannier90.wout at the end of the run.

Plot the wannier functions and get Hamiltonian file

```
$ sftp citimacguest@tejo.unican.es
$ mget workshop/myname/*xsf
$ quit
$ xcrysden
```

If you have time prepare the input for MODELMAKER.PY

```
$ sftp citimacguest@tejo.unican.es
$ get wannier90_tb.dat
```

Step 0 - Check the folders and files

1. Go to systemref This folder contains the most symmetric calculation and closest to what the reference state should be

   ```
   $ cd systemref
   $ vi silicon.info
   ...
   ```

   - Pay attention to the format of the files
   - Check particularly silicon.info and silicon.wann
   - silicon.info keywords used to converge this calculation
   - silicon.wann definition of the reference state

2. Go to training

   ```
   $ cd ../training
   $ vi silicon_m01.info
   ...
   ```

   - Note that each calculation has its own name

     ```
     $ ls
     ```

   - Note that there is no .wann files not reference state!

3. Check input.mmaker

   ```
   $ cd ..
   $ vi input.mmaker
   ```

Step 0 - input.mmaker

- model_file silicon.xml

  the name of the output file

- !lattice_file silicon_lat.xml

  if previous lattice model add electron over its file (commented)

- cutoff word word=energy/atom

  parameters selected only on energy criterion or all interactions from atom with 1 interaction below threshold

- cutoff_h value

  the cutoff for the 1-electron hamiltonian

- cutoff_pos value

  add element if $|\vec{r}_{ab}|$ is large enough even if γ_{ab} below cutoff_h

- cutoff_ee value

  the cutoff for the 2-electron hamiltonian

- ksampling value value value

  the internal k-sampling used to generate occupations, etc.

- neutral_model word word=yes/no

  Create model so that atoms are neutral (recommended)

Step 1 - The general quality of the model

Goal: Set the right value for cutoff_h

1. Edit the input file:

   ```
   $ vi input.mmaker
   ```

   Set the value of cutoff_h to some large value (0.5 should do)

2. Run modelmaker:

   ```
   $ python $SCALEUPHOME/scripts/modelmaker.py 1
   ```

   Plot: black line→FP, green→best possible SP

   How good is the representation of the bands?

3. Edit the input file again and lower cutoff_h (0.1?)

   ```
   $ vi input.mmaker
   $ python $SCALEUPHOME/scripts/modelmaker.py 1
   ```

4. Iterate until satisfied with bands

   We look for insulator with indirect gap

   How many terms include each value of cutoff_h?

Step 2 - The occupations

Goal: Decide occupations of reference state

1. Run modelmaker:

   ```
   $ python $SCALEUPHOME/scripts/modelmaker.py 2
   ```

   Look at the occupation of each WF Does it look ok?

2. Edit the input file:

   ```
   $ vi input.mmaker
   ```

   Make the reciprocal grip finer and rerun modelmaker

3. Iterate until reasonable

   - Is the ground state in SP going to be the reference state
   - What is the most reasonable reference state?

4. Edit the systemref/silicon.wann file to set the reference

   ```
   $ vi systemref/silicon.wann
   ```

Step 3 - Selecting the terms

Goal: Decide which terms enter in the model

1. Select cutoff_ee and run modelmaker:

   ```
   $ vi input.mmaker
   $ python $SCALEUPHOME/scripts/modelmaker.py 3 ¿ out_3
   ```

   Look at the output file:

   ```
   $ vi out_3
   ```

2. Check the one-electron terms:

   The Gamma representatives:

   The one-electron degrees of freedom Are the ones expected?

   Very small values?→ check cutoff_pos

3. Check the two-electron terms:

   Printing _u_list.dat vs interactions with largest 2e variations

   Compare with D(h_U) in Printing table with all data

Is it reasonable?

- YES Proceed to fit (step 4)
- NO Check cutoff_ee or edit _u_list.dat/_gamma_list.dat

```
$ vi _u_list.dat
```

Step 4 - The fit

Goal: Generate the model file

1. Run the fit:

```
$ unbuffer python $SCALEUPHOME/scripts/modelmaker.py 4 —
tee -a out_4
```

or

```
$ python $SCALEUPHOME/scripts/modelmaker.py 4
```

This can take a long time!

2. Something failed? Check output in run_scaleup directory
3. Look at the parameters: Do they make sense? if NOT:

Edit the parameter files and enter better initial guesses:

```
$ vi _u_list.dat
```

and rerun:

```
$ python $SCALEUPHOME/scripts/modelmaker.py 4
```

4. Problems still? Choose better terms/change training set:

Many problems come from having too many parameters!

5. At the end of the fit always check the error of the parameters

Large error indicates a weak parameter!

Practical session summary

- 1-Wannier functions

  Calculate the WFs of silicon using LDA and decide the adequate windows and projections

- 2-TB model of silicon

  Go to 1_silicon_tb and obtain a TB model of silicon using modelmaker

  While step 4 runs move to the next example

- 2-Hubbard model of silicon

  Look at training set to obtain U and check selected U terms

```
$ python $SCALEUPHOME/scripts/modelmaker.py mymode
```