Introduction to Wannier90 and Tight-binding fitting

Muhammad Yusrul Hanna

Kelompok Riset Teori Materi Kuantum

Pusat Riset Fisika Kuantum BRIN

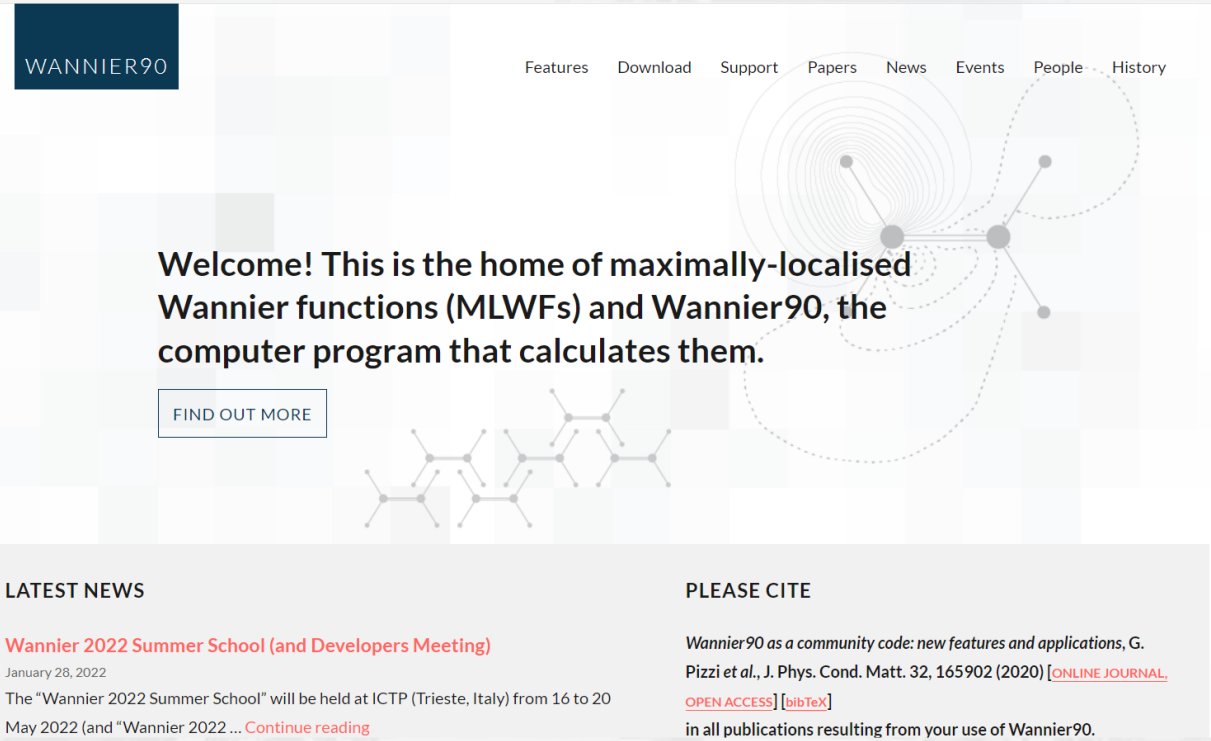
Workshop on Starter Packages for

COMPUTATIONAL CONDENSED MATTER PHYSICS RESEARCH

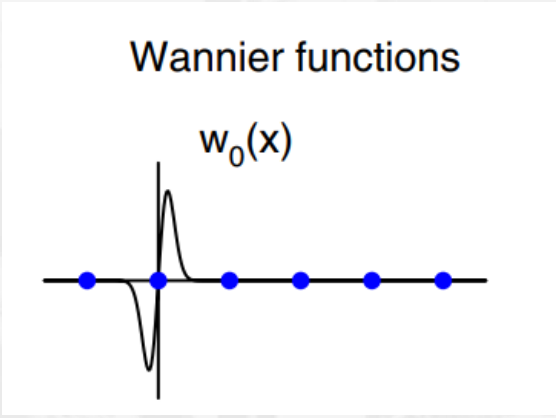
15 April 2023

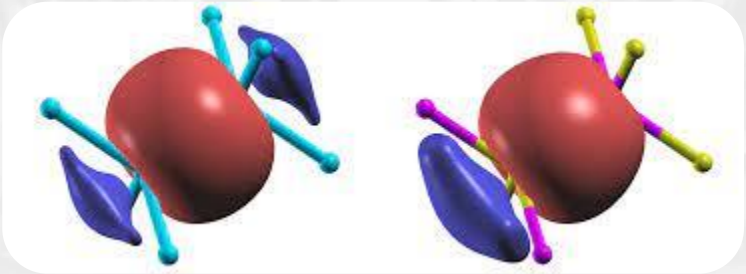
Overview of Wannier90

http://www.wannier.org/



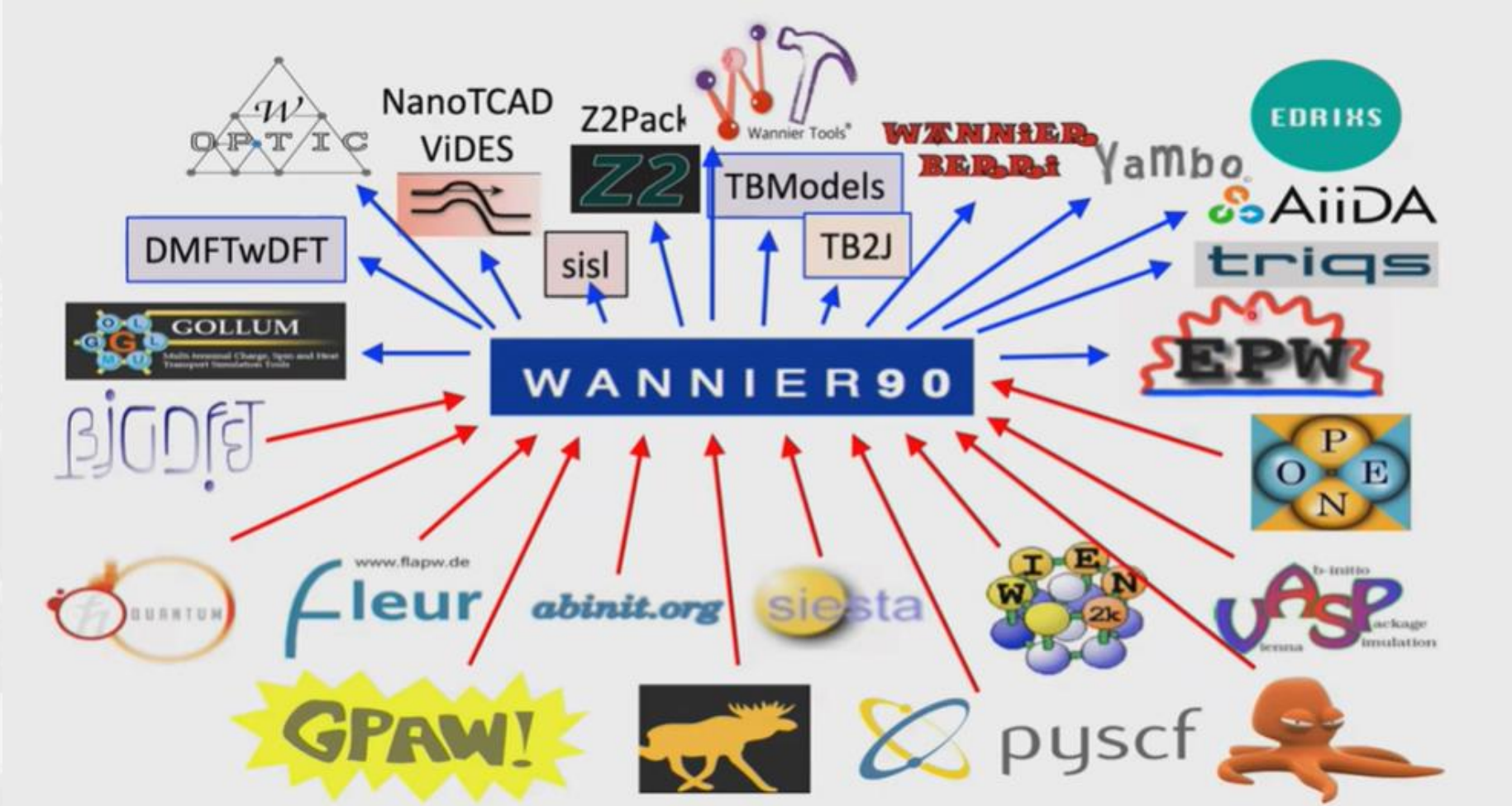
Wannier90 is an open-source code for generating **maximally-localized Wannier functions** and using them to compute advanced electronic properties of materials with high efficiency and accuracy. It exploits the **real-space localisation of WFs** to obtain many spectral and Fermi-surface properties at high-resolution in the Brillouin zone (so-called Wannier interpolation)





Marzari, Mostofi, Yates, Souza & Vanderbilt, RMP **84**, 1419 (2012)

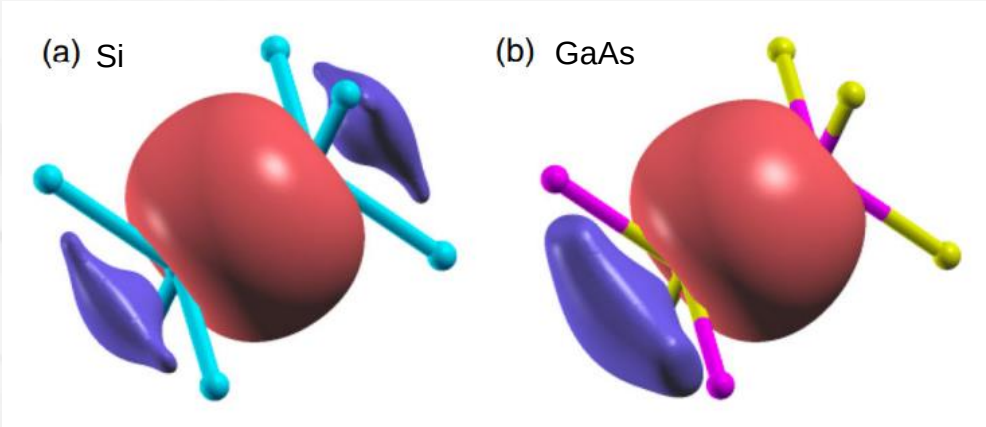
Overview of Wannier90

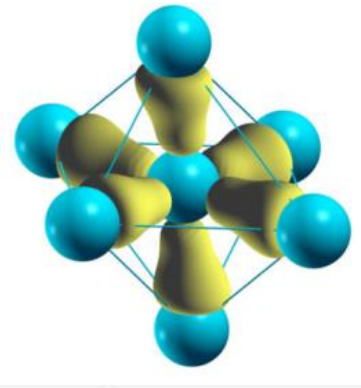


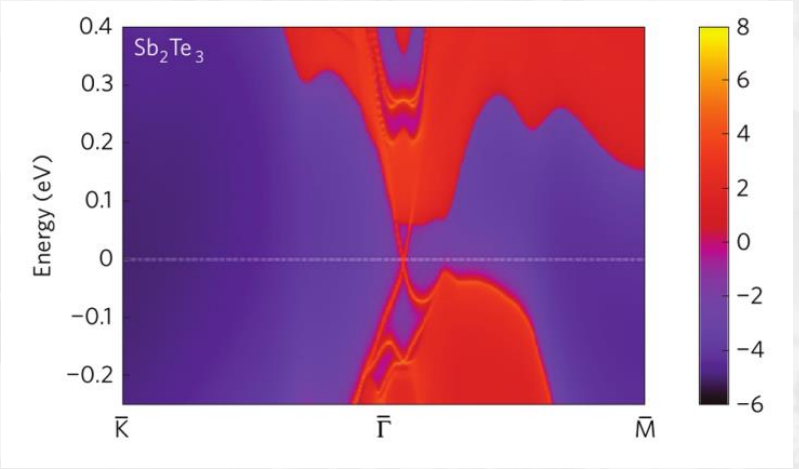
Arash Mostofi talk: https://www.youtube.com/watch?v=8sxAr0Rtp2k&t=3216s

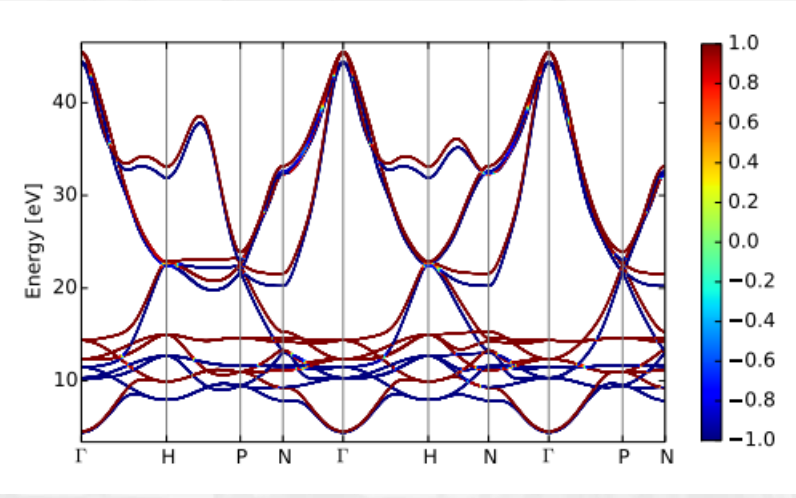
Wannier90 Features

- Localized, real space picture of electronic structures
- Interpretation of chemical bonding
- Polarization and orbital magnetization
- Efficient Brillouin zone interpolation and integration
- Berry phase properties
- Electronic transport
- Topological properties
- Constructing minimal basis-sets





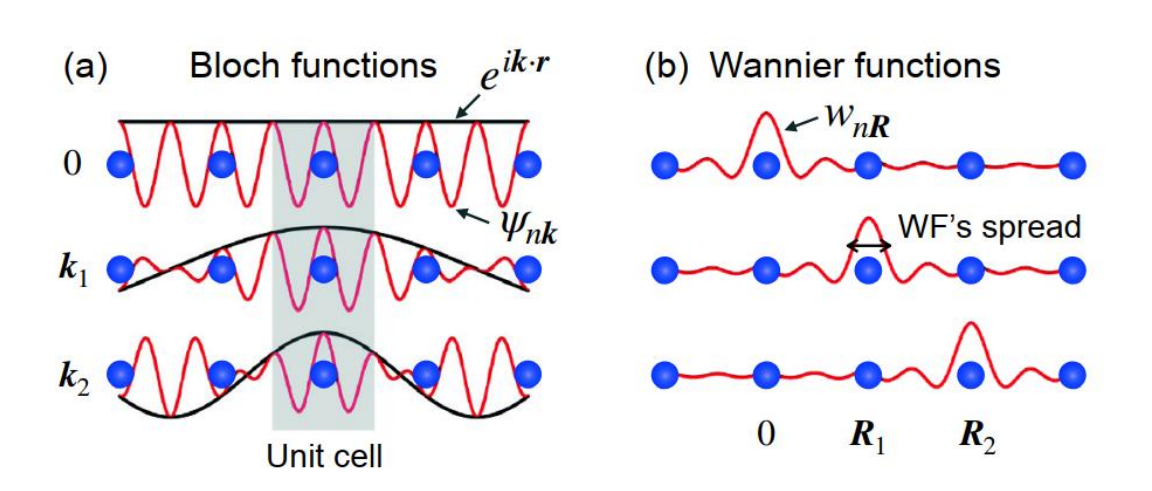




Arash Mostofi talk: https://www.youtube.com/watch?v=8sxAr0Rtp2k&t=3216s

Wannier functions

Wannier functions (WFs) : a smaller set of Fourier-transformed Bloch functions from DFT calculations that are localized in real space.



Oscillating & delocalized

Localized

Bloch state of energy band

$$\psi_{n\boldsymbol{k}}(\boldsymbol{r}) = \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}\cdot\boldsymbol{R}} w_{n\boldsymbol{R}}(\boldsymbol{r})$$

Wannier function

Inverse Fourier transformation

Wannier function

$$w_{n\boldsymbol{R}}(\boldsymbol{r}) = \frac{1}{N_{\boldsymbol{k}}} \sum_{\boldsymbol{k}} e^{-i\boldsymbol{k}\cdot\boldsymbol{R}} \psi_{n\boldsymbol{k}}(\boldsymbol{r})$$

periodicity as the atomic structure

Wannier functions not unique → Bloch function are not unique

$$w_{n\boldsymbol{R}}(\boldsymbol{r}) = \frac{1}{N_{\boldsymbol{k}}} \sum_{\boldsymbol{k}} e^{-i\boldsymbol{k}\cdot\boldsymbol{R}} \left[\sum_{m=1}^{N_e} U_{mn}^{\boldsymbol{k}} \psi_{m\boldsymbol{k}}(\boldsymbol{r}) \right]$$

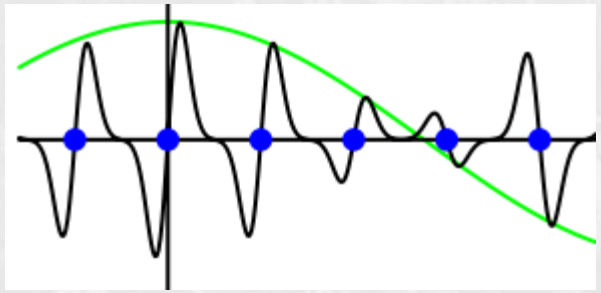
How to choose U_{mn}^{k} to obtain MLWFs?

Hung, Nugraha & Saito, *Quantum ESPRESSO course for solid state physics* (2022)

Maximally-localized Wannier functions (MLWFs)

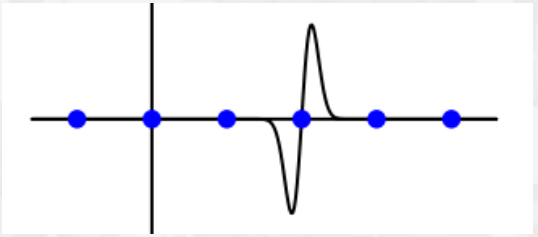
MLWFs: effective method to minimizing the mean square spread (Ω) of the Wannier Functions to choose $U_{mn}^{\boldsymbol{k}}$

$$w_{n\boldsymbol{R}}(\boldsymbol{r}) = \frac{1}{N_{\boldsymbol{k}}} \sum_{\boldsymbol{k}} e^{-i\boldsymbol{k}\cdot\boldsymbol{R}} \left[\sum_{m=1}^{N_e} U_{mn}^{\boldsymbol{k}} \psi_{m\boldsymbol{k}}(\boldsymbol{r}) \right]$$



$$\Omega = \sum_{n} \left[\langle r^2 \rangle_n - \langle \boldsymbol{r} \rangle_n^2 \right]$$

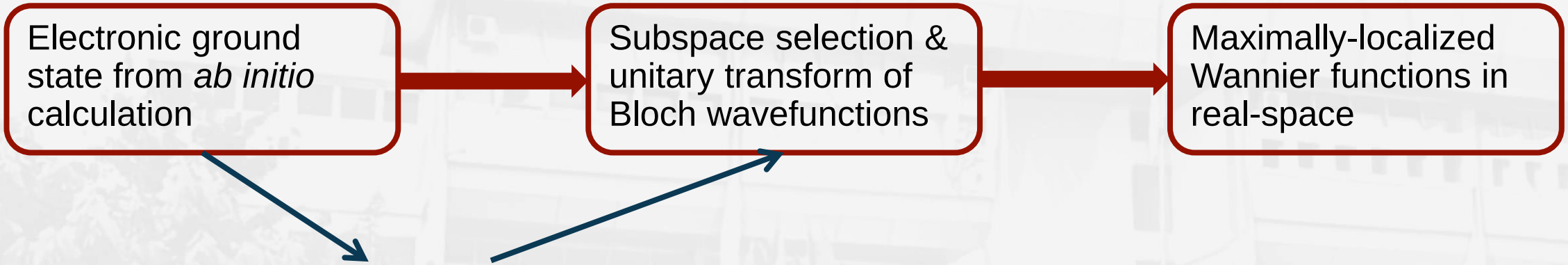
the center and the second moment of the average for WF at $\boldsymbol{R}$ = 0



Minimize spread functional Ω w.r.t unitary transformations $U_{mn}^{\boldsymbol{k}}$

Marzari & Vanderbilt, PRB 56, 12847 (1997); Souza, Marzaro & Vanderbilt, PRB 65, 035109 (2001)

Maximally-localized Wannier functions (MLWFs)



Express Ω in terms of overlaps of the periodic parts of the Bloch functions

$$M_{mn}^{\boldsymbol{k},\boldsymbol{b}} = \langle u_{m\boldsymbol{k}} | u_{n,\boldsymbol{k}+\boldsymbol{b}} \rangle$$ Overlap matrix element

Initial U may be constructed from a set of localized projection functions $g(\boldsymbol{r})$

$$A_{mn}^{\boldsymbol{k}} = \langle \psi_{m\boldsymbol{k}} | g_n \rangle$$ Projected matrix elements

$g(\boldsymbol{r})$ are typically localized atom-or bond-centered orbitals: s,p, d, f, sp, sp^2, sp^3, sp^3d and sp^3d^2

Arash Mostofi talk: https://www.youtube.com/watch?v=8sxAr0Rtp2k&t=3216s

Tight-binding model and Wannier interpolation

The tight-binding model is a technique used to calculate the electronic band structure of materials. It is based on the principle that the wave function of an electron in a crystal can be represented by combining atomic orbitals found on different lattice sites.

Wannier interpolation is a method used to create tight-binding models for larger systems, such as semi-infinite surfaces. This is accomplished by using maximally localized Wannier functions obtained from various systems.

Effective model Hamiltonian in real space for the tight-binding

Hopping parameters

$$\mathcal{H} = \sum_{\boldsymbol{R}\boldsymbol{R}'} \sum_{nn'} t_{nn'}\left(\boldsymbol{R}, \boldsymbol{R}'\right) \left|w_{n\boldsymbol{R}}\right\rangle \left\langle w_{n'\boldsymbol{R}'}\right|$$

$$\mathcal{H}_{nn'}(\boldsymbol{q}) = \sum_{\boldsymbol{R}} e^{i\boldsymbol{q}\cdot\boldsymbol{R}} t_{nn'}(\boldsymbol{R})$$

Kohn-Sham eigenvalues

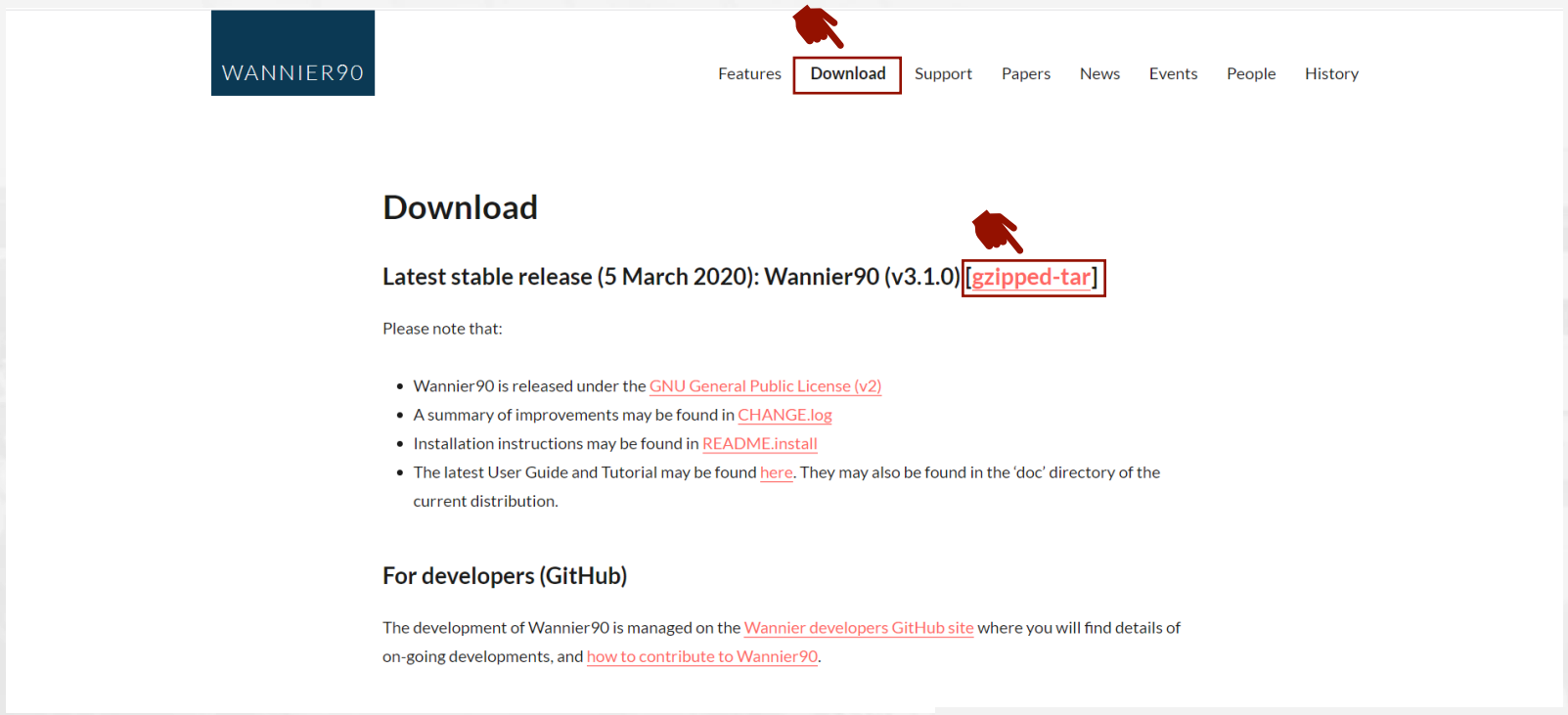
$$= \sum_{\boldsymbol{R}} e^{i\boldsymbol{q}\cdot\boldsymbol{R}} \left(\frac{1}{N_{\boldsymbol{k}}} \sum_{\boldsymbol{k}} e^{-i\boldsymbol{k}\cdot\boldsymbol{R}} \left[\left(U^{\boldsymbol{k}}\right)^{\dagger} \epsilon(\boldsymbol{k}) U^{\boldsymbol{k}} \right]_{nn'} \right)$$

Diagonalizing $\mathcal{H}_{nn'}(\boldsymbol{q})$ → The energy dispersion $\epsilon(\boldsymbol{q})$

q-grid can be much denser than k-grid

Wannier interpolation

Wannier90 installation




```
$ sudo apt install build-essential gfortran libopenmpi-dev
$ sudo apt install libblas-dev liblapack-dev libfftw3-dev
```

```
$ wget https://github.com/wannier-developers/wannier90/archive/v3.1.0.tar.gz
$ tar -zxvf v3.1.0.tar.gz
```

```
$ cd wannier90-3.1.0
$ cp config /make/inc.gfort make inc
$ make all
```

```
$ export PATH = $PATH:/home/username/wannier90-3.1.0/
```

Edi Suprayoga tutorial: https://www.youtube.com/watch?v=AZVQGCixdgk&t=6s

Wannier90 files

Generated from pwscf (pw2wannier90.x)

<seedname>.mmn → Bloch overlaps

<seedname>.amn → Projections

<seedname>.eig → eigenvalues

Wannier90 input file

<seedname>.win

How to calculate …

QUANTUMESPRESSO

Ground state calculation

Obtain Bloch states on a uniform grid

Ab initio

Wannier90.x -pp <seedname>

Generate list of required overlaps (<seedname>.nnkp)

*pw2wannier90.x < *.pw2wan > *pw2wan.out*

- Overlap $\langle u_{n\boldsymbol{k}} | u_{n(\boldsymbol{k}+\boldsymbol{b})} \rangle$ between Bloch states (<seedname>.mmn)
- The projection for starting guess (<seedname>.amn)

Wannier90.x <seedname>

Compute MLWFs

Wannier90 input

```
##########################################
# W90 MODULES
##########################################
num_bands          = 16          ! number of DFT bands
num_wann           = 5           ! number of Wannier bands

! ------------------------ !
! dis_froz is set to [-3 eV:+3 eV]
! with respect to Fermi level
! EF = -1.6773  eV
! ------------------------ !
dis_froz_max       = 1.3227
dis_froz_min       = -4.6773

! ------------------------ !
! dis_window is set to [-20 eV:+20 eV]
! with respect to Fermi level
! EF = -1.6773 eV
! ------------------------ !
#dis_win_min        = -23.9808
#dis_win_max        = 16.3192

dis_num_iter       = 5000
num_iter           = 000
num_print_cycles   = 100
dis_mix_ratio      = 1.0
iprint             = 2
##########################################
```

```
##########################################
# EXTRACT TIGHT-BINDING PARAMETER
##########################################
write_hr = .true.
write_xyz = .true.
use_ws_distance  = .true.
```

```
##########################################
# SYSTEM
##########################################
begin unit_cell_cart
Ang
    2.463924    0.000000    0.000000
   -1.231962    2.133821    0.000000
    0.000000    0.000000   15.000000
end unit_cell_cart

begin atoms_frac
C        0.333333    0.666667    0.500000
C        0.666667    0.333333    0.500000
end atoms_frac
```

Wannier90 input

```
##########################################
# PROJECTION
##########################################
begin projections
C: pz
f= 0.5, 0.5, 0.5: s
f= 0.5, 0.0, 0.5: s
f= 0.0, 0.5, 0.5: s
end projections
```

```
##########################################
# PLOT BANDSTRUCTURE
##########################################
bands_plot = .true.
bands_num_points = 200

begin kpoint_path
G   0.00000   0.00000   0.00000       K   0.33333   0.33333   0.00000
K   0.33333   0.33333   0.00000       M   0.50000   0.00000   0.00000
M   0.50000   0.00000   0.00000       G   0.00000   0.00000   0.00000
end kpoint_path
```

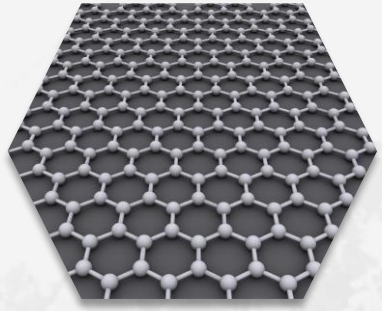
```
##########################################
# K-POINTS
##########################################
mp_grid =  12   12   1

begin kpoints
  0.00000000  0.00000000  0.00000000
  0.00000000  0.08333333  0.00000000
  0.00000000  0.16666667  0.00000000
  0.00000000  0.25000000  0.00000000
  0.00000000  0.33333333  0.00000000
  0.00000000  0.41666667  0.00000000
  0.00000000  0.50000000  0.00000000
  0.00000000  0.58333333  0.00000000
  0.00000000  0.66666667  0.00000000
  0.00000000  0.75000000  0.00000000
  0.00000000  0.83333333  0.00000000
  0.00000000  0.91666667  0.00000000
  0.08333333  0.00000000  0.00000000
  0.08333333  0.08333333  0.00000000
  ...
  ...
  ...
  ...
end kpoints
```

/home/username/wannier90-3.1.0/utility/kmesh.pl 12 12 1 → nscf calculation

/home/username /wannier90-3.1.0/utility/kmesh.pl 12 12 1 w → seedname.win

Application-Graphene



For more detail calculation, please visit :https://github.com/myhanna/wannier90-tutorial-unhas

$vi gr_hr.dat

```
...
   0   0   0   5   1    0.000000   -0.000000
   0   0   0   1   2   -2.865473    0.000000
   0   0   0   2   2   -1.328247   -0.000000
...
   0   0   0   5   5  -12.361820    0.000000
   0   1   0   1   1    0.252175   -0.000000
   0   1   0   2   1    0.026367   -0.000000
...
```

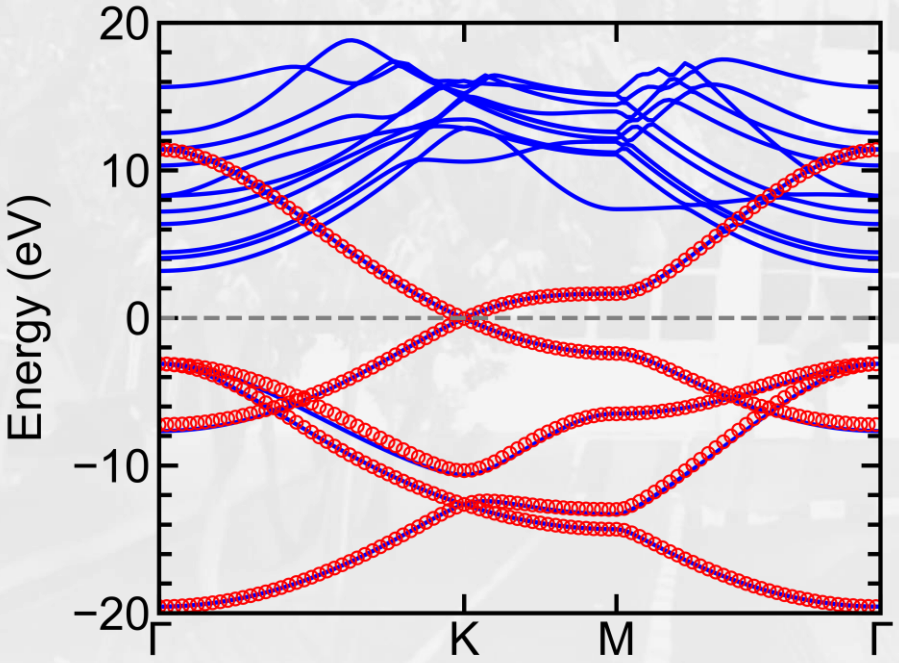
Translation vector of unit cell

WF indices

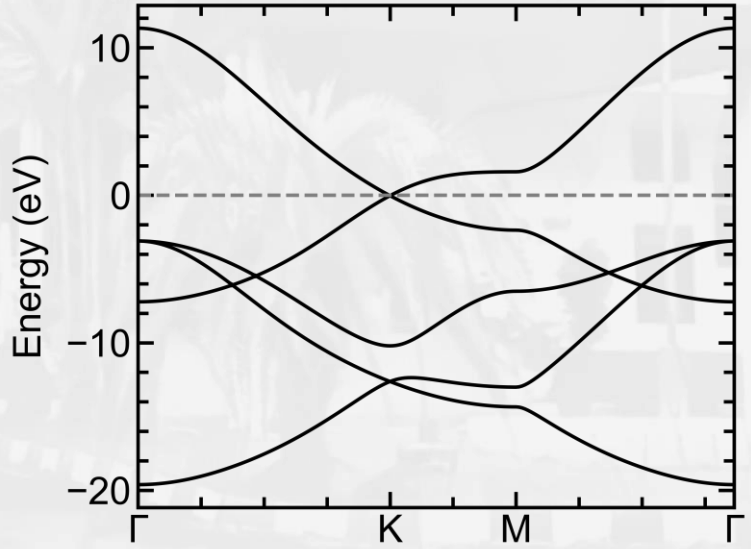
Real and imaginary part of hopping parameters

t_{11} for T=(0,0,0) = -2.86 eV
t_{12} for T=(0,1,0) = 0.25 eV

DFT and wannier interpolation



Tight-binding model

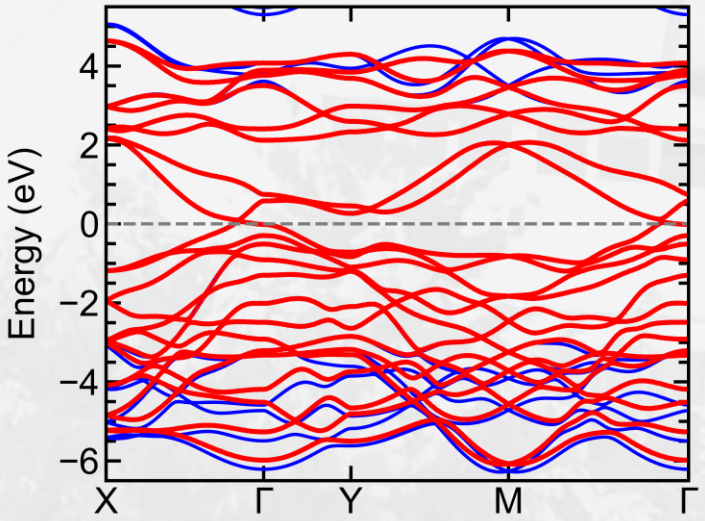


Construct tight-binding parameter using **pythtb** package

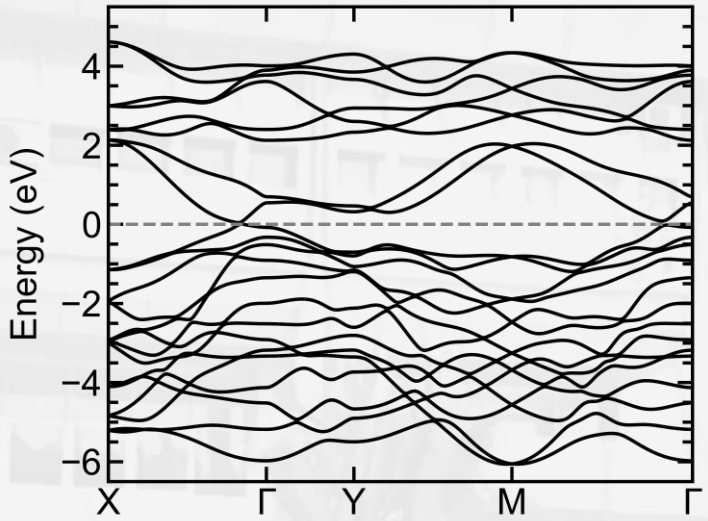
$pip install pythtb

[Bonus] Edge States 1T'-MoS2

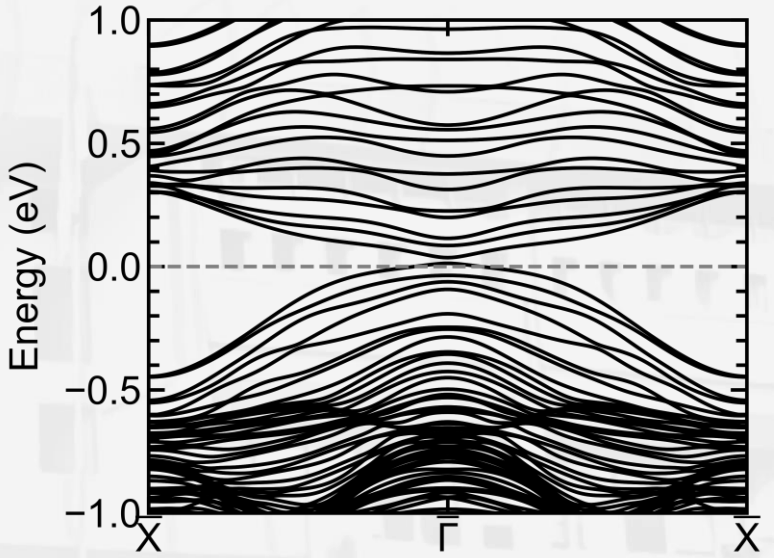
DFT and Wannier interpolation



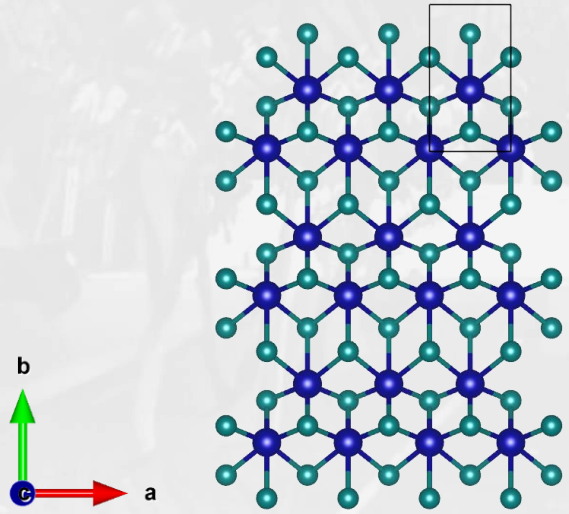
Tight-binding model

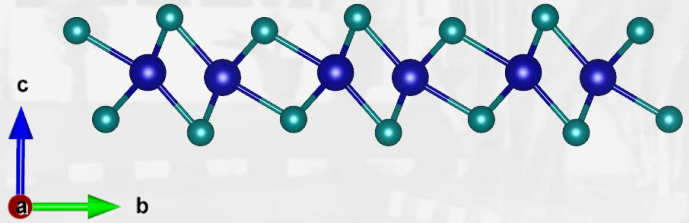


Edge state



Nslab = 20 ; (100) surface





In topological materials, **edge states** are **localized** electronic states that exist at the **edges** of a sample. These states are protected by symmetry, making them robust against structural defects or disorder at the interface between two media with different topological properties.

For more detail calculation, please visit :https://github.com/myhanna/wannier90-tutorial-unhas

@created by S.A.W

Thank You

for your attention

ありがとうございました

If you have any questions about Wannier90, please do not hesitate to contact me at muha207@brin.go.id.