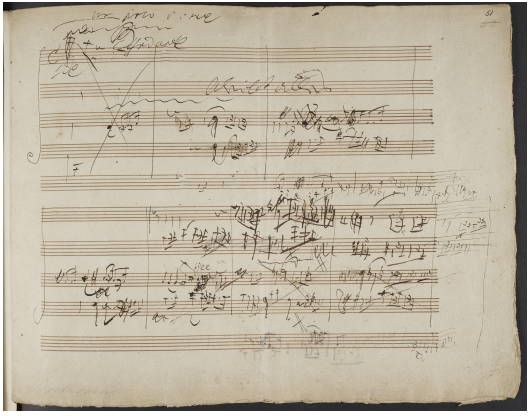
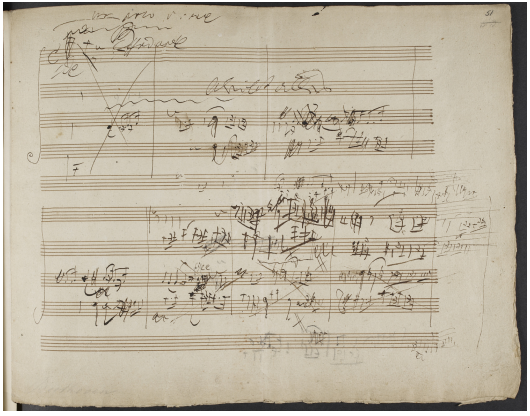
String Quartet No. 14 - Adagio ma non troppo e molto espressivo



click here for video

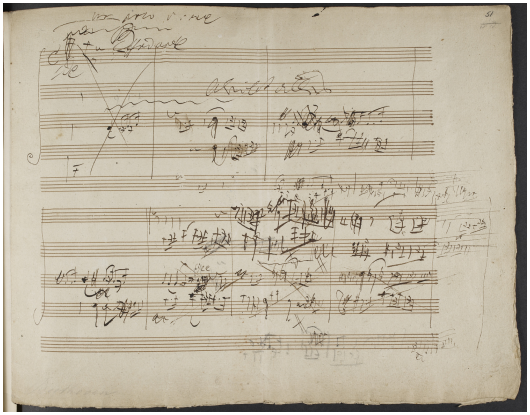
String Quartet No. 14 - Adagio ma non troppo e molto espressivo



click here for video

"indecipherable, uncorrected horrors" – Spohr

String Quartet No. 14 - Adagio ma non troppo e molto espressivo



click here for video

"indecipherable, uncorrected horrors" – Spohr
"After this, what is left for us to write?" – Schubert

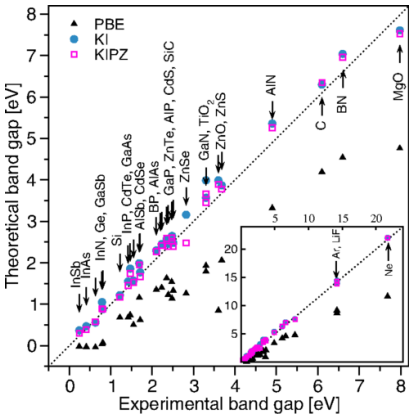
Towards black-box Koopmans band structures

or: getting lost down a pseudopotential-generation rabbit hole

Edward Linscott | PSI | THEOS Group Meeting | ? January 2024 |

Swiss National Science Foundation
MARVEL

Koopmans functionals give accurate band structures



Mean absolute error (eV) across prototypical semiconductors and insulators

	PBE	G_0W_0	KI	KIPZ	QSG$\tilde{W}$
E_{gap}	2.54	0.56	0.27	0.22	0.18
IP	1.09	0.39	0.19	0.21	0.49

Koopmans functionals give accurate band structures

	PBE	G_0W_0[1]	scG$\tilde{W}$[2]	KI@[PBE,MLWFs]	KIPZ@PBE	exp[3]
E_g	0.49	1.06	1.14	1.16	1.15	1.17
$\Gamma_{1v} \rightarrow \Gamma_{25'v}$	11.97	12.04		11.97	12.09	12.5 $\pm$0.6
$X_{1v} \rightarrow \Gamma_{25'v}$	7.82			7.82		7.75
$X_{4v} \rightarrow \Gamma_{25'v}$	2.85	2.99		2.85	2.86	2.90
$L_{2'v} \rightarrow \Gamma_{25'v}$	9.63	9.79		9.63	9.74	9.3 $\pm$0.4
$L_{1v} \rightarrow \Gamma_{25'v}$	6.98	7.18		6.98	7.04	6.8 $\pm$0.2
$L_{3'v} \rightarrow \Gamma_{25'v}$	1.19	1.27		1.19		1.2 $\pm$0.2
$\Gamma_{25'v} \rightarrow \Gamma_{15c}$	2.48	3.29		3.17	3.20	3.35$\pm$0.01
$\Gamma_{25'v} \rightarrow \Gamma_{2'c}$	3.28	4.02		3.95	3.95	4.15$\pm$0.05
$\Gamma_{25'v} \rightarrow X_{1c}$	0.62	1.38		1.28	1.31	1.13
$\Gamma_{25'v} \rightarrow L_{1c}$	1.45	2.21		2.12	2.13	2.04$\pm$0.06
$\Gamma_{25'v} \rightarrow L_{3c}$	3.24	4.18		3.91	3.94	3.9 $\pm$0.1
MSE	0.35	0.02		0.01	0.03	
MAE	0.44	0.21		0.14	0.17	

[1] M. Shishkin et al. *Phys. Rev. Lett.* 99.24 (2007), 246403 for E_g and M. S. Hybertsen et al. *Phys. Rev. B* 34.8 (1986), 5390 for the transitions;

[2] M. Shishkin et al. *Phys. Rev. B* 75.23 (2007), 235102.

[3] O. Madelung. *Semiconductors*. 3rd ed. Berlin: Springer-Verlag, 2004.

Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition

Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition
- is orbital-density-dependent

Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition
- is orbital-density-dependent
- relies on localization

Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition
- is orbital-density-dependent
- relies on localization
- requires the ab initio calculation of screening parameters

Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition
- is orbital-density-dependent
- relies on localization
- requires the ab initio calculation of screening parameters

In order to evaluate this functional, one must...

- initialize a set of variational orbitals

Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition
- is orbital-density-dependent
- relies on localization
- requires the ab initio calculation of screening parameters

In order to evaluate this functional, one must...

- initialize a set of variational orbitals
- calculate the screening parameters $\{\alpha_i\}$

Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

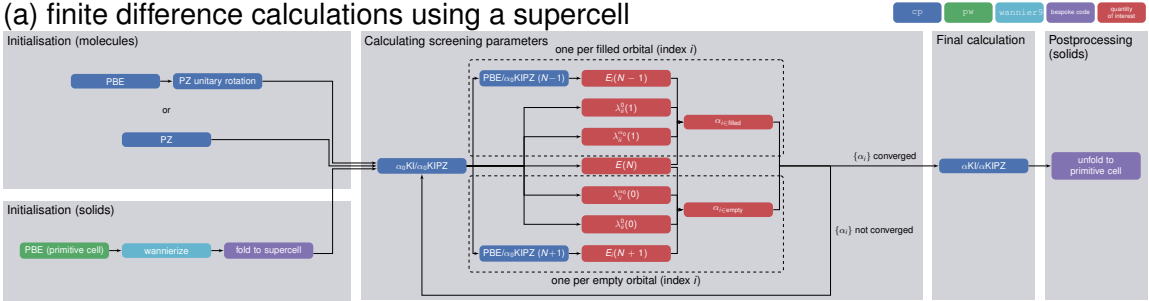
- a correction to DFT that enforces a generalized piecewise linearity condition
- is orbital-density-dependent
- relies on localization
- requires the ab initio calculation of screening parameters

In order to evaluate this functional, one must...

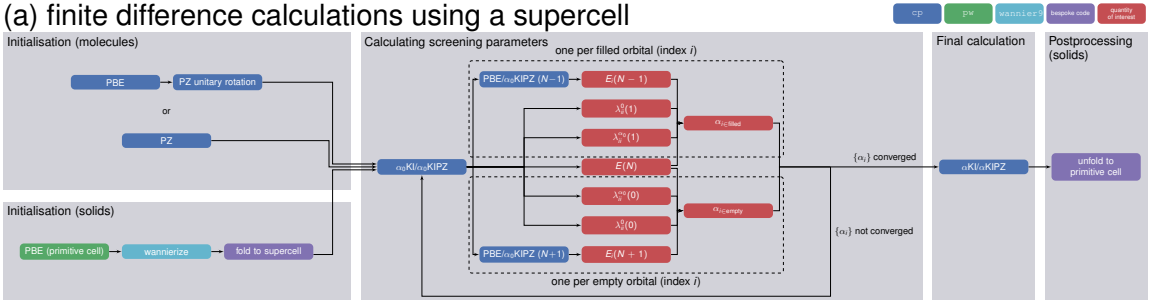
- initialize a set of variational orbitals
- calculate the screening parameters $\{\alpha_i\}$
- construct and diagonalize the Hamiltonian

Workflows

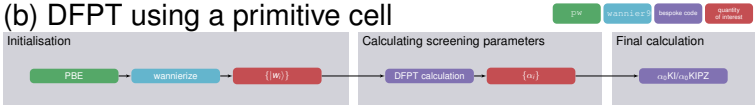
(a) finite difference calculations using a supercell



(a) finite difference calculations using a supercell

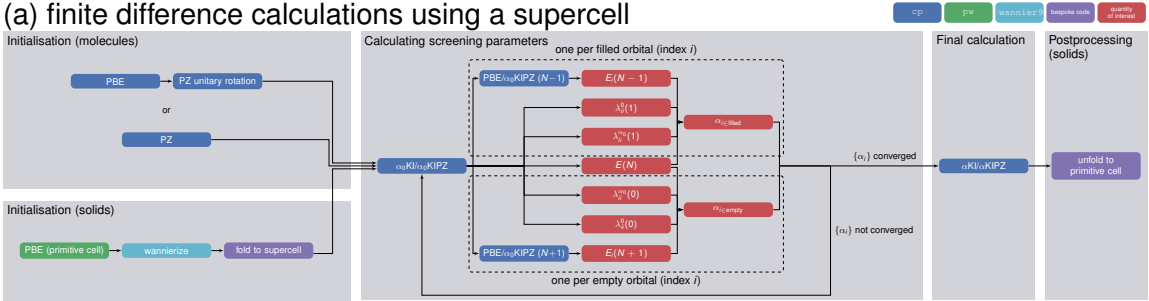


(b) DFPT using a primitive cell

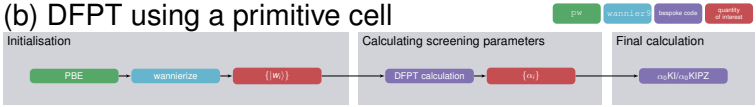


Workflows

(a) finite difference calculations using a supercell



(b) DFPT using a primitive cell



All implemented in Koopmans

Workflows

What still stands in our way? Take the example of silicon:

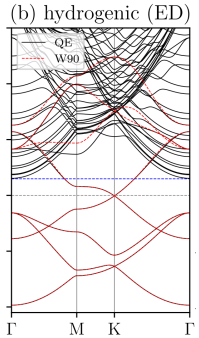
```
{
  "workflow": {
    "task": "singlepoint",
    "functional": "ki",
    "base_functional": "lda",
    "method": "dfpt",
    "pseudo_library": "pseudo_dojo_standard"},
  "atoms": {
    "cell_parameters": {"periodic": true, "ibrav": 2, "celldms": {"1": 10.2622}},
    "atomic_positions": {
       "units": "crystal",
       "positions": [["Si", 0.00, 0.00, 0.00], ["Si", 0.25, 0.25, 0.25]]}},
  "kpoints": {"grid": [8, 8, 8]},
  "calculator_parameters": {
    "ecutwfc": 60.0,
    "pw": {"nbnd": 20},
    "w90": {
      "projections": [[{"fsite": [0.25, 0.25, 0.25], "ang_mtm": "sp3"}],
                      [{"fsite":[ 0.25, 0.25, 0.25], "ang_mtm": "sp3"}]],
      "dis_froz_max": 10.6,
      "dis_win_max": 16.9}}}
```

Automating Wannierization

One very manual step: Wannierization. Can we automate this?

Automating Wannierization

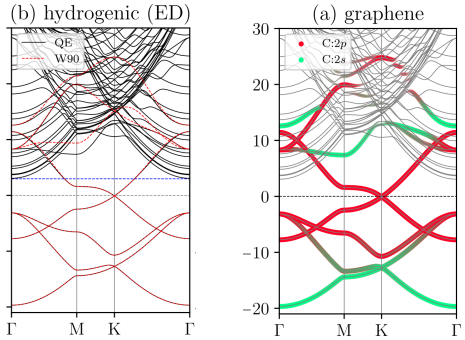
One very manual step: Wannierization. Can we automate this?



J. Qiao et al. *npj Comput Mater* 9 (2023), 208

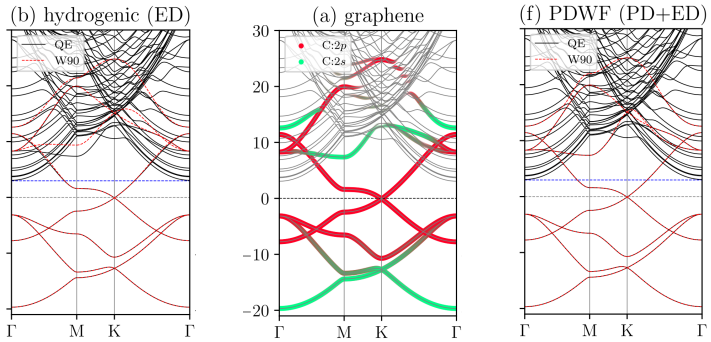
Automating Wannierization

One very manual step: Wannierization. Can we automate this?



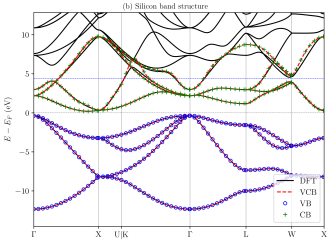
Automating Wannierization

One very manual step: Wannierization. Can we automate this?



Automating Wannierization

We separate target manifolds via parallel transport to obtain separate occupied and empty manifolds



Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability
- they serve as our initial guesses for the Wannier functions

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF
Li $1s^2$ $2s^1$

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF
Li $1s^2\ 2s^1 \rightarrow$ 2 PAOs (1s, 2s)

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF
Li $1s^2$ $2s^1$ $\rightarrow$ 2 PAOs (1s, 2s)
F $[1s^2]$ $2s^2$ $2p^5$

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF
Li $1s^2\ 2s^1 \rightarrow$ 2 PAOs (1s, 2s)
F $[1s^2]\ 2s^2\ 2p^5 \rightarrow$ 4 PAOs (1s, 2s, $2p_x$, $2p_y$, $2p_z$)

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF
Li $1s^2\ 2s^1 \rightarrow$ 2 PAOs (1s, 2s)
F $[1s^2]\ 2s^2\ 2p^5 \rightarrow$ 4 PAOs (1s, 2s, $2p_x$, $2p_y$, $2p_z$)
6 Wannier functions for a system with 10 electrons

Automating Wannierization

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF
Li $1s^2\ 2s^1 \rightarrow$ 2 PAOs (1s, 2s)
F $[1s^2]\ 2s^2\ 2p^5 \rightarrow$ 4 PAOs (1s, 2s, $2p_x$, $2p_y$, $2p_z$)
6 Wannier functions for a system with 10 electrons = 5 occupied bands + only 1 unoccupied band!

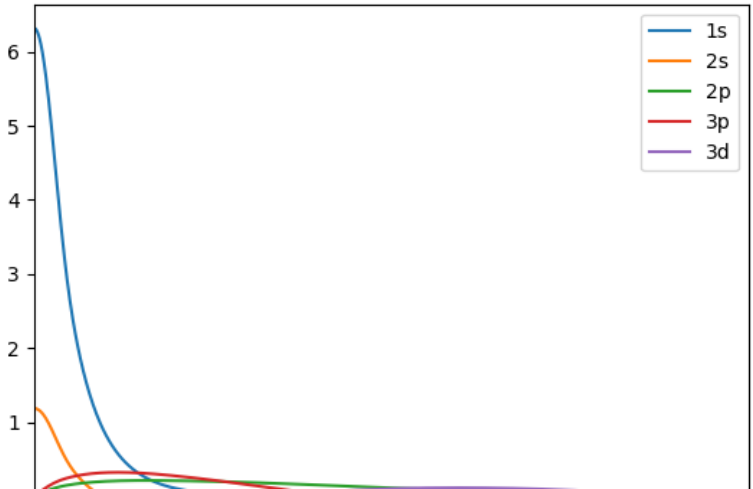
Automating Wannierization

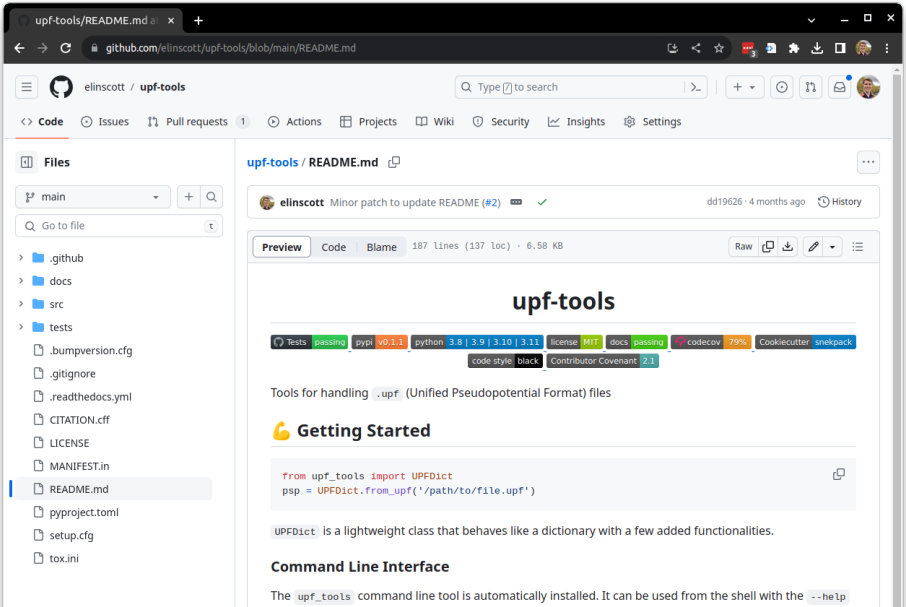
Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability $\rightarrow$ we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions $\rightarrow$ the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF
Li $1s^2\ 2s^1 \rightarrow$ 2 PAOs (1s, 2s)
F $[1s^2]\ 2s^2\ 2p^5 \rightarrow$ 4 PAOs (1s, 2s, $2p_x$, $2p_y$, $2p_z$)
6 Wannier functions for a system with 10 electrons = 5 occupied bands + only 1 unoccupied band!
If we want more Wannier functions, we're gonna need ~~a bigger boat~~ more PAOs...

Existing strategy: use the PAOs provided by OpenMX

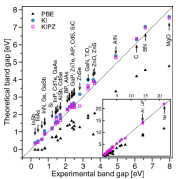


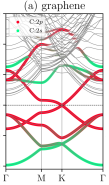

upf-tools/README.md
github.com/elinscott/upf-tools/blob/main/README.md
elinscott / upf-tools
Type / to search
Code
Issues
Pull requests 1
Actions
Projects
Wiki
Security
Insights
Settings
Files
main
Go to file t
.github
docs
src
tests
.bumpversion.cfg
.gitignore
.readthedocs.yml
CITATION.cff
LICENSE
MANIFEST.in
README.md
pyproject.toml
setup.cfg
tox.ini
upf-tools / README.md
elinscott Minor patch to update README (#2)
dd19626 · 4 months ago
History
Preview
Code
Blame
187 lines (137 loc) · 6.58 KB
Raw
upf-tools
Tests passing
pypi v0.1.1
python 3.8 | 3.9 | 3.10 | 3.11
license MIT
docs passing
codecov 79%
Cookiecutter snekpack
code style black
Contributor Covenant 2.1
Tools for handling .upf (Unified Pseudopotential Format) files
Getting Started
from upf_tools import UPFDict
psp = UPFDict.from_upf('/path/to/file.upf')
UPFDict is a lightweight class that behaves like a dictionary with a few added functionalities.
Command Line Interface
The upf_tools command line tool is automatically installed. It can be used from the shell with the --help

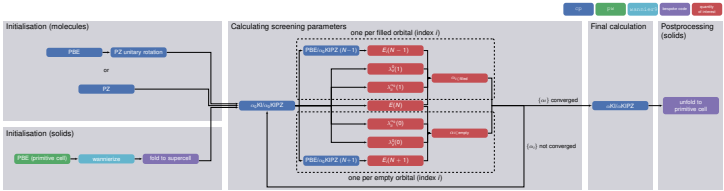
- Most of the infrastructure was from the `cookiecutter` I used (see my July 2022 GM)
- Publish your code! `upf-tools` was discovered by Marnik and is now used in `AiiDA`
- Also have unmerged tools for `oncvpsp` input and output files as well as Junfeng's custom `.dat` projector files (perhaps `oncvpsp-tools` Need to decide on the right home for these...

Take home messages

Edward Linscott
PSI | 14/15







- Koopmans functionals yield band structures with comparable accuracy to state-of-the-art GW
- the release of `koopmans` means non-experts can now use Koopmans functionals in their own research
- work is ongoing to automate the Wannierization bottleneck

Acknowledgements

Nicola Marzari

Nicola Colonna

Riccardo De Gennaro

Junfeng Qiao

Swiss National Science Foundation

MARVEL

NATIONAL CENTRE OF COMPETENCE IN RESEARCH

slides available at github/elinscott-talks and on the THEOS wiki