Imperial College
London

June 10, 2022

Introduction to VASP

a piratical guide from an user's perspective

Chengcheng Xiao

Imperial College London

About me

My name is Chengcheng Xiao

I'm a 3rd year PhD student working with Prof. Arash Mostofi.

I've been an avid VASP user for about 6 years now.

My email: `cx219@ic.ac.uk`

Introduction

The **Vienna Ab initio Simulation Package** (VASP), is a package for performing ab-initio quantum mechanical calculations using the projector augmented wave (PAW) method, and plane wave basis.

The basic methodology adopted by VASP is **density functional theory (DFT)**, but the code also allows use of post-DFT corrections such as **hybrid functionals** mixing DFT and Hartree–Fock exchange (e.g. HSE, PBE0 or B3LYP), many-body perturbation theory (the GW method) and dynamical electronic correlations within the random phase approximation.

Introduction [comparison]

Comparing to other DFT codes that uses PW basis set (e.g. QuantumESPRESSO), VASP is (by my own humble opinions):

- Faster for larger systems (especially for hybrids).
- Easier to use (both in terms of numerical stability and user friendliness and user documentations).
- It's the most widely used DFT code in the world [1] and most likely the answer to the problem you are having can be found online.
- its very accurate (according to the delta test of DFT codes `https://molmod.ugent.be/deltacodesdft`).

[1] Don't quote me on this! As a matter of fact, don't quote me on any of these...

Imperial College
London

Introduction [comparison]

At the same time, when comparing to open source DFT codes like QuantumESPRESSO, VASP has it's down sides[2]:

- It puts more stuff under-the-hood (again, less parameters to play with).
- You won't be getting the newest features as the code is more of a "downstream" code.
- modifying the code to add new capabilities can be more extraneous since the in-line documentation is not done very well (some of them are in German...)

[2]Again, don't quote on any of these...

Imperial College London

Outline

In this short talk, I will give;

- A brief introduction to PAW formalism.
- A quck installation guide.
- A quick introduction to the inputs.
- A brief introduction to some post-processing/interfacing codes.

Before we start

Before we start, I need to stress that **VASP is a commercial code**. This means you'll need to have a valid license to "possess" it's source code (and provided pseudopotential files) and the code should, in principle, only be compiled on machines that have been permitted a prior by the VASP company.

All publication that contains results calculated by VASP should cite relevant papers. A list can be found: 🔗 here.

Imperial College London

A brief intro to PAW

VASP works within the PAW formalism and comes with a suite of well-tested, highly accurate PAW datasets.

This is both for historical reasons and for practical reasons (in some cases PAW can be more accurate and the transferability can be sometimes be better comparing to other methods) and in theory PAW method can achieve all-electron accuracy.

Imperial College London

A brief intro to PAW [Transformation]

The core of the PAW formalism is the linear transformation of the all-electron wavefunctons:

$$|\psi_{nk}\rangle = |\tilde{\psi}_{nk}\rangle + \sum_i (|\phi_i\rangle - |\tilde{\phi}_i\rangle) \langle \tilde{p}_i|\tilde{\psi}_{nk}\rangle \tag{1}$$

where $|\psi_{nk}\rangle$ is the all-electron (AE) wavefunctions with the wiggles in side the core region and $|\tilde{\psi}_{nk}\rangle$ is the pseudo (PS) wavefunctions which only differ from $|\psi_{nk}\rangle$ inside the core region (i.e. inside the PAW sphere). $|\phi_{nk}\rangle$ and $|\tilde{\phi}_{nk}\rangle$ are AE and PS partial waves and $\tilde{p}_i$ is the so-called β projector.

In a sense, the PAW method is an all-electron method.

Imperial College London

A brief intro to PAW [Transformation]

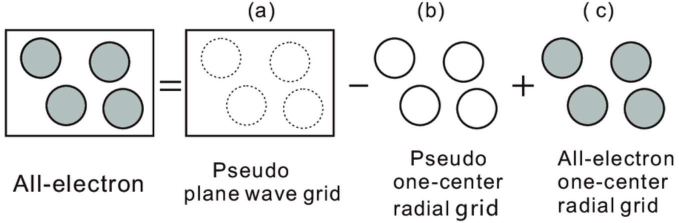

Figure 1: An illustration of additive augmentation in the PAW method. (a) Pseudized quantities are defined on a regular plane wave grid in the entire space. (b) Pseudo wave functions are reconstructed inside spheres and the corresponding one-center terms are subtracted. (c) The all-electron wave functions are reconstructed and the corresponding one-center energies are added

Imperial College London

A brief intro to PAW [Quantities]

The variational quantity $|\tilde{\psi}_{nk}\rangle$ is expanded by a set of plane waves:

$$\langle \mathbf{r} \mid \tilde{\psi}_{n\mathbf{k}} \rangle = \frac{1}{\Omega^{1/2}} \sum_{\mathbf{G}} C_{n\mathbf{k}\mathbf{G}} e^{i(\mathbf{G}+\mathbf{k})\cdot\mathbf{r}} \tag{2}$$

The AE and PS partial waves on atom α:

$$\langle \mathbf{r} \mid \phi_{\alpha} \rangle = \frac{1}{|\mathbf{r}-\mathbf{R}_{\alpha}|} u_{l_{\alpha}\varepsilon_{\alpha}}\left(|\mathbf{r}-\mathbf{R}_{\alpha}|\right) Y_{l_{\alpha}m_{\alpha}}\left(\mathbf{r}-\mathbf{R}_{\alpha}\right) \tag{3}$$

$$\langle \mathbf{r} \mid \tilde{\phi}_{\alpha} \rangle = \frac{1}{|\mathbf{r}-\mathbf{R}_{\alpha}|} \tilde{u}_{l_{\alpha}\varepsilon_{\alpha}}\left(|\mathbf{r}-\mathbf{R}_{\alpha}|\right) Y_{l_{\alpha}m_{\alpha}}\left(\mathbf{r}-\mathbf{R}_{\alpha}\right) \tag{4}$$

note that only the radial part of them differ, so that we only need to pseudoize the radial wavefunctions.

Imperial College London

A brief intro to PAW [Quantities]

The projector $|\tilde{p}_i\rangle$ only exist inside the PAW sphere and is dual to the PS partial wave:

$$\int_0^{r_c^{PAW}} \tilde{p}_i(r)^* \tilde{\phi}_j(r) dr = \delta_{ij} \tag{5}$$

Since the projector only exist in the PAW sphere (very localized) and the variational quantity $\tilde{\psi}$ is projected onto them, the real-space projection can introduce some numerical inaccuracies. To prevent this, **the projection is usually done in rec space**. (the tag `LREAL` controls this)

Imperial College
London

A brief intro to PAW (Quantities)

The projector is constructed by the usual ultra-soft pseudopotential (USPP) way. Using a set of l dependent local potentials that map smoothly to the AE potentials outside the PAW sphere, the projectors can be written as:

$$\begin{aligned} |\chi\rangle &= (\epsilon - \hat{T} - V_{\text{loc}})\,|\tilde{\phi}\rangle \\ B &= \langle\tilde{\phi}|\chi\rangle \\ |\tilde{p}\rangle &= (B^{-1})\,|\chi\rangle \end{aligned} \tag{6}$$

This can also be generalized into multiple projectors to improve the scattering properties of the PPs.

Imperial College London

A brief intro to PAW [example]

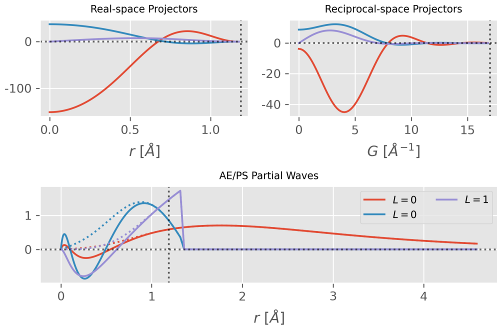

Figure 2: PAW dataset for sodium. Upper: Projectors $|\tilde{p}\rangle$ in real and reciprocal space. Lower panel, AE/PS partial waves (solid/dashed lines)[3]

[3]this plot is made using 🔗VASPbandunfolding

Imperial College
London

A brief intro to PAW (charge density)

Since the PS partial waves does not conserve the norm of the AE wavefunction, the charge density has to be augmented:

$$n(\mathbf{r}) = \widetilde{n}(\mathbf{r}) - \widetilde{n}^1(\mathbf{r}) + n^1(\mathbf{r})$$

$$\widetilde{n}(\mathbf{r}) = \left\langle \widetilde{\psi}_{n\mathbf{k}} \mid \mathbf{r} \right\rangle \left\langle \mathbf{r} \mid \widetilde{\psi}_{m\mathbf{k}} \right\rangle$$

$$\widetilde{n}^1(\mathbf{r}) = \sum_{\alpha,\beta} \widetilde{\phi}^*_\alpha(\mathbf{r}) \widetilde{\phi}_\beta(\mathbf{r}) \left\langle \widetilde{\psi}_{n\mathbf{k}} \mid \tilde{p}_\alpha \right\rangle \left\langle \tilde{p}_\beta \mid \widetilde{\psi}_{m\mathbf{k}} \right\rangle$$

$$n^1(\mathbf{r}) = \sum_{\alpha,\beta} \phi^*_\alpha(\mathbf{r}) \phi_\beta(\mathbf{r}) \left\langle \widetilde{\psi}_{n\mathbf{k}} \mid \tilde{p}_\alpha \right\rangle \left\langle \tilde{p}_\beta \mid \widetilde{\psi}_{m\mathbf{k}} \right\rangle$$

Imperial College London

A brief intro to PAW (charge density)

$$\begin{aligned} n(\mathbf{r}) &= \widetilde{\psi}_{n\mathbf{k}}(\mathbf{r})\widetilde{\psi}_{m\mathbf{k}}(\mathbf{r}) \\ &\quad + \sum_{\alpha,\beta}\left[\phi^*_\alpha(\mathbf{r})\phi_\beta(\mathbf{r}) - \widetilde{\phi}^*_\alpha(\mathbf{r})\widetilde{\phi}_\beta(\mathbf{r})\right]\langle\widetilde{\psi}_{n\mathbf{k}}|\widetilde{p}_\alpha\rangle\langle\widetilde{p}_\beta|\widetilde{\psi}_{m\mathbf{k}}\rangle \\ &= \widetilde{\psi}_{n\mathbf{k}}(r)\widetilde{\psi}_{m\mathbf{k}}(r) + \sum_{\alpha,\beta} Q_{\alpha,\beta}\left\langle\widetilde{\psi}_{n\mathbf{k}} \mid \tilde{p}_\alpha\right\rangle\left\langle\tilde{p}_\beta \mid \widetilde{\psi}_{m\mathbf{k}}\right\rangle \end{aligned} \tag{7}$$

where,

$$Q_{\alpha,\beta}(\mathbf{r}) = \phi^*_\alpha(\mathbf{r})\phi_\beta(\mathbf{r}) - \widetilde{\phi}^*_\alpha(\mathbf{r})\widetilde{\phi}_\beta(\mathbf{r}). \tag{8}$$

And we can define the overlap operator $\hat{S}$:

$$\hat{S} = \hat{1} + \sum_{\alpha,\beta}\left[\int_0^{r_c^{\text{PAW}}} Q_{\alpha,\beta}(\mathbf{r})dr\right]|\tilde{p}_\alpha\rangle\langle\tilde{p}_\beta| \tag{9}$$

Imperial College London

A brief intro to PAW (operator)

Similarly, we can write any local operator $\tilde{A}$ as:

$$\tilde{A} = \hat{A} + \sum_{\alpha,\beta} \left[\int_0^{r_c^{\text{PAW}}} \phi_\alpha^*(\mathbf{r}) \hat{A} \phi_\beta(\mathbf{r}) dr - \int_0^{r_c^{\text{PAW}}} \widetilde{\phi}_\alpha^*(\mathbf{r}) \hat{A} \widetilde{\phi}_\beta^*(\mathbf{r}) \right] |\tilde{p}_\alpha\rangle \langle \tilde{p}_\beta| \tag{10}$$

Imperial College
London

A brief intro to PAW (more to read)

some more reading materials:

- http://www.physics.metu.edu.tr/~hande/teaching/741-lectures/lecture-08.pdf
- http://www.tcm.phy.cam.ac.uk/castep/CASTEP_talks_06/pickard1.pdf
- https://www.tcm.phy.cam.ac.uk/castep/CASTEP_talks_07/carlsson2.pdf
- https://www.theoretical-physics.net/dev/quantum/paw.html
- https://www.vasp.at/wiki/index.php/Projector-augmented-wave_formalism

How to install VASP

The VASP source code is distributed in the following style:

```
               vasp.x.x.x (root directory)
                        |
  ------------------------------------------------
 |        |        |         |          |         |
arch     bin     build      src     testsuite   tools
                             |
                       -------------
                      |      |      |
                     lib   parser  fftlib
```

Imperial College
London

How to install VASP

`Arch` directory contains the exemplary `makefile.include.xxx` needed for compilation for different type of machines. We need to copy one of them to `root directory` as `makefile.include`.

Usually, on `X86` machines, one can just use the `makefile.include.linux_intel` and load the proper Intel compilers. For example, on CX1:

```
mdule purge
module load intel-suite/2017.6
module load mpi/intel-2019
```

Imperial College London

How to install VASP

After putting the makefile.include in the root dir, to compile VASP just type:

```
make all
```

and three executable named vasp_std, vasp_gam, vasp_ncl will appear in the bin directory.

---

vasp_std is the standard executable.

vasp_gam is a "gamma-only" version which speeds up the calculation with symmetry (real wavefunctions).

vasp_ncl is the "non-collinear" version which can be used for calculations that requires spin-orbit coupling.

Imperial College London

How to install VASP (advanced)

There are a lot of additional functionalities that can be activated while compiling VASP.

Most of these features can be activated by adding additional compiler flags in `makefile.include`'s `CPP_OPTIONS` and setting the path to corresponding external libraries and header files by modifying `LLIBS` and `INCS`.

For more information on how to do this for specific packages, consult this webpage `https://www.vasp.at/wiki/index.php/Makefile.include`.

Imperial College London

How to install VASP (GPU version)

VASP has been ported to **the GPU with openACC**. However, as of now, it's only available for Nvidia cards. The speed up using GPU version is pretty substantial especially when doing Hybrids.

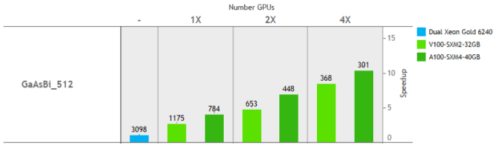

Figure 3: GPU speed up comparing to Xeon Gold 6240 18 cores @ 2.6G base clock

To install the GPU version, please consult: 🔗our group WIKI or 🔗this official document.

How to use VASP

VASP requires four input files:

- INCAR : includes all the control parameters.
- KPOINT : includes all information about the k-points settings.
- POSCAR : includes the structures.
- POTCAR : includes the potential files.

Some typical output files:

- OUTCAR: includes all analytical outputs.
- CHGCAR: Charge density.
- WAVECAR: Wave functions. (Binary file)

Imperial College London

Inputs [POSCAR]

POSCAR file is structured as follow:

```
Cubic BN          ! COMMENT
   3.57           ! SCALING FACTOR FOR LATTICE VEC
 0.0 0.5 0.5      ! LATTICE VECTOR 1
 0.5 0.0 0.5      ! LATTICE VECTOR 2
 0.5 0.5 0.0      ! LATTICE VECTOR 3
   B N            ! ATOM TYPES
   1 1            ! NUMBER OF ATOMS FOR EACH TYPE
Direct            ! POSITION TYPE: DIRECT -> CRYSTAL
 0.00 0.00 0.00 ! POSITION OF THE FIRST ATOM (B)
 0.25 0.25 0.25 ! POSITION OF THE SECOND ATOM (N)
```

Imperial College
London

Inputs [POSCAR]

To fix atoms position during relaxation:

```
Cubic BN                    ! COMMENT
   3.57                     ! SCALING FACTOR FOR LATTICE VEC
 0.0 0.5 0.5                ! LATTICE VECTOR 1
 0.5 0.0 0.5                ! LATTICE VECTOR 2
 0.5 0.5 0.0                ! LATTICE VECTOR 3
   B N                      ! ATOM TYPES
   1 1                      ! NUMBER OF ATOMS FOR EACH TYPE
Selective dynamics          ! TURN ON TO FIXING ATOMS
Direct                      ! POSITION TYPE: DIRECT -> CRYSTAL
 0.00 0.00 0.00 F F F! POSITION ALONG A B fixed
 0.25 0.25 0.25 T T T!
```

Imperial College London

Inputs [POTCAR]

The POTCAR file contains the PAW datasets of each element types that appears in the POSCAR file. One can generate the POTCAR file simply by congregating POTCAR files of each elements using:

```
cat $VASP_PP_PATH/element1/POTCAR \
    $VASP_PP_PATH/element2/POTCAR >> POTCAR
```

Imperial College
London

Inputs [KPOINTS]

KPOINTS file has the following structure (for uniform MP-mesh):

```
K-point              ! COMMENT LINE
0                    ! NOT USED HERE
G                    ! GAMMA CENTERED MP MESH
14     7      1      ! NUMBER OF K-POINTS ALONG EACH AXIS
```

Imperial College London

Inputs [KPOINTS]

Line mode for band structures:

```
K-point                 ! COMMENT LINE
30                      ! 30 POINTS BETWEEN NODES
Line-mode               ! TURN ON LINE MODE
rec                     ! THE FOLLOWING ARE IN REC COORD.
0    0    0             ! GAMMA
0.5  0    0             ! X
0.5  0.5  0             ! K
```

Imperial College London

Inputs [INCAR]

INCAR controls everything from the electronic minimization algorithm to what files to print and what type of calculations to perform.

Suggestion

Build a generic INCAR file and modify upon for each use case.

For more details, please visit:
`https://www.vasp.at/wiki/index.php/INCAR`

Or visit our group wiki:
`https://friendly-broccoli-22e4d939.pages.github.io/code_usage/vasp/geometry/`

Imperial College London

Inputs [INCAR]

A typical INCAR contains:

```
SYSTEM = system name
ISTART = 0          ! start fresh
ENCUT  = 500        ! energy cutoff
NELM   = 200        ! max number of electronic steps
NSW    = 0          ! max number of ionic steps
IBRION = -1         ! ionic relax algo
ISMEAR = 0          ! smearing method
SIGMA  = 0.01       ! smearing width
ALGO   = N          ! algo for electronic minimization
```

Imperial College London

Running VASP

Running VASP is simple since it doesn't support any command line arguments.

```
mpiexec -n 24 /PATH/TO/VASP/vasp_std | tee log
```

In this command, the `stdout` is redirected(copied) to `log` file and 24 MPI processes are lunched to run VASP in parallel.

Imperial College
London

How to use VASP [workflow]

The basic workflow is:

- Relax the structure: 🔗LINK.
  - cell relax: `ISIF`, beware of the Pulay stress.
  - fix axis during cell relax: 🔗my repo
  - fix atomic positions: `selective dynamics`
- SCF calculation:🔗LINK.
  - generate charge density: `LCHARG`
  - generate wavefunctions: `LWAVE`
- Calculate band structures using calculated charge densities: 🔗LINK.
  - read-in charge density: `LCHARGE`
  - use Gaussian smearing: `ISMAER`, `SIGMA`

Imperial College London

How to use VASP [advanced]

- To stop calculation at the next step: use `STOPCAR`
- To use manual occupation: `ISMEAR=-2` and `FERWE`,`FERWO`.
- To instruct the code not to do electronic minimization: set `ALGO=None`
- Currently the DFPT method implemented in VASP is γ only.
- setting `LREAL=Auto` can cause substantial speed up, but only use this for large systems.

Post-processing tools

- Phono(3)py:`http://phonopy.github.io/phonopy/`
- VTST: `https://theory.cm.utexas.edu/vtsttools/`
- VASPsol: `https://github.com/henniggroup/VASPsol`
- TRIQS: `https://triqs.github.io/dft_tools/latest/guide/conv_vasp.html`
- PeriodicNBO: `https://schmidt.chem.wisc.edu/nbosoftware`
- SSAdNDP: `http://ion.chem.usu.edu/~boldyrev/ssadndp.php`
- VASPKIT: `https://vaspkit.com/`
- Wannier90: `http://www.wannier.org/`
- py4vasp: `https://www.vasp.at/py4vasp/latest/`

Imperial College London

More to read

- https://www.nsc.liu.se/support/Events/VASP_workshop_2020/seminar2.pdf
- https://www.vasp.at/wiki/index.php/Category:Tutorials
- https://docs.archer2.ac.uk/research-software/vasp/
- https://portal.supercomputing.wales/wp-content/uploads/2018/06/Lab_Worksheet_VASP_SCW_SLURM.pdf
- https://www.nersc.gov/assets/Uploads/pap134.pdf
- https://www.nsc.liu.se/~pla/blog/2014/02/21/deltacodes/

Imperial College
London

The End

```
 ------------------------------------------
|                                          |
|  ADVICE TO THIS USER RUNNING 'VASP/VAMP' |
|  (HEAR YOUR MASTER'S VOICE ...):         |
|                                          |
| Follow me on Github: @Chengcheng-Xiao    |
| Read my blog:                            |
|    https://chengcheng-xiao.github.io/    |
|                                          |
 ------------------------------------------
```

Imperial College
London

The End

Find these slides: