



# Workshop and International Conference on Electronic Structure Theory of Emergent Spin Orbit Driven Phenomenon

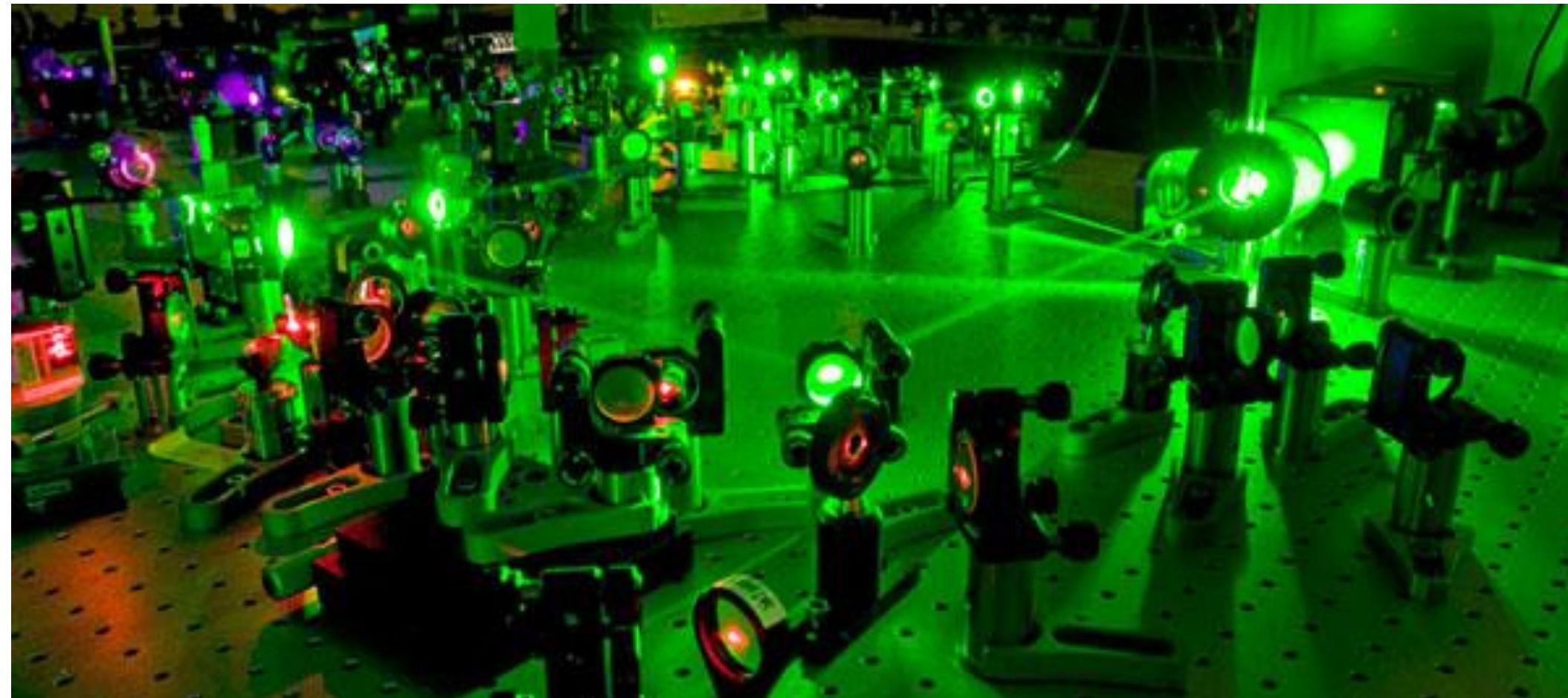
## Tutorial 1: Basics of DFT calculations - self consistent loops and geometry optimization

**Afternoon session: 2:30 p.m. – 4 p.m.**

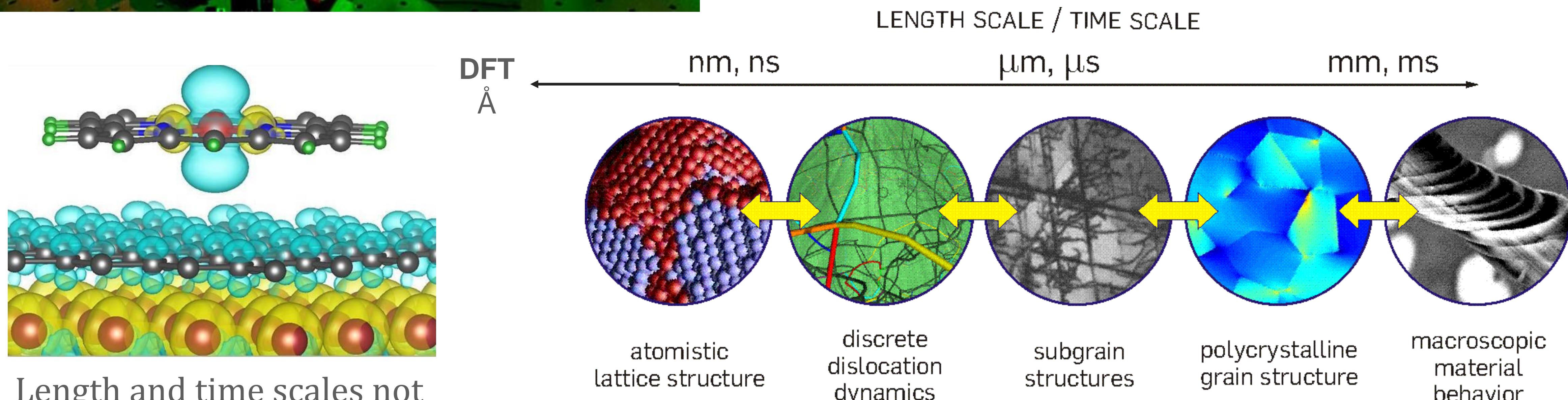
Dipanwita Bhattacharjee

# Why atomistic first-principles calculations?

ESTESODP



- Nanoscale phenomenon: small length scales ( $\sim 10^{-9}\text{m}$ ) and small time scales ( $\sim 10^{-15}\text{s}$ ) are difficult to observe in experiments
- Real experiments are often time-consuming and expensive

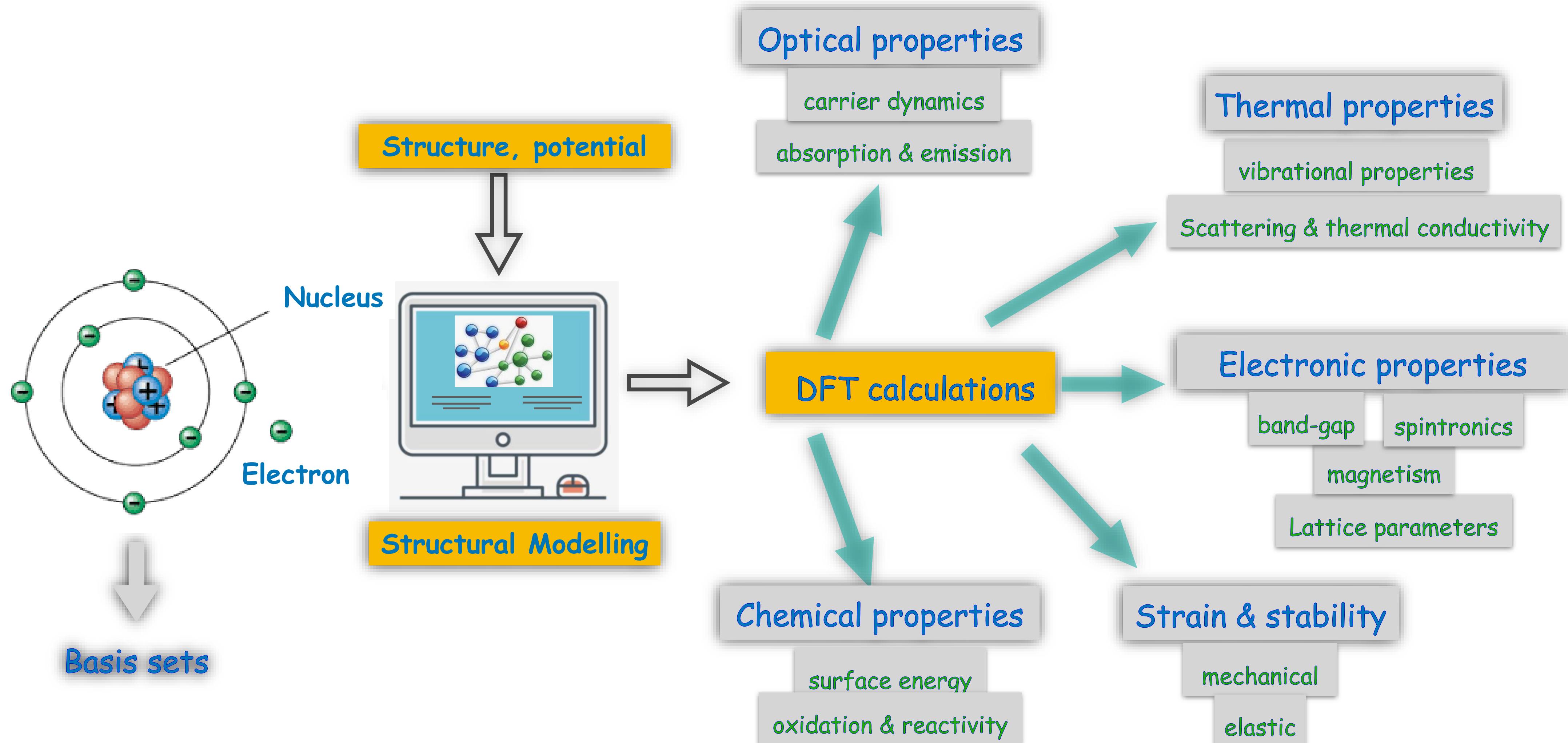


Length and time scales not describable by continuum approaches

A route from the microscopic (electrons, atoms and their interactions) to the macroscopic (equation of state, transport coefficients)

# Applications

ESTESODP

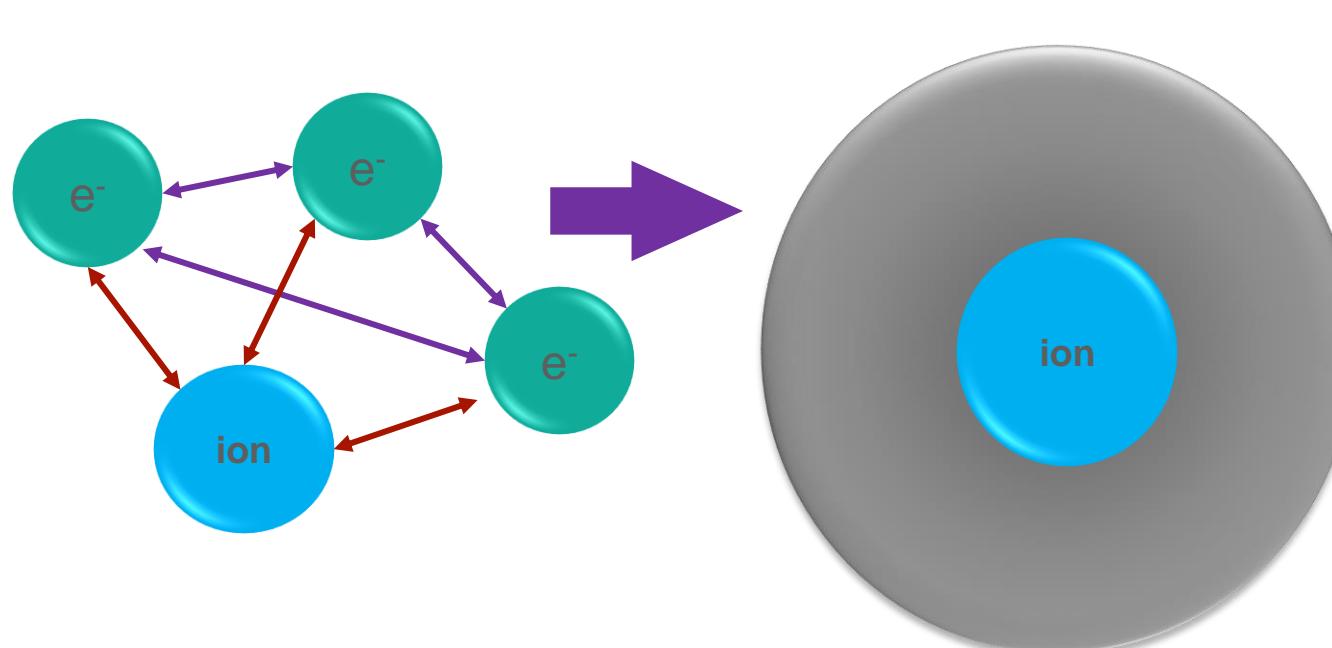


# Density Functional Theory (DFT)

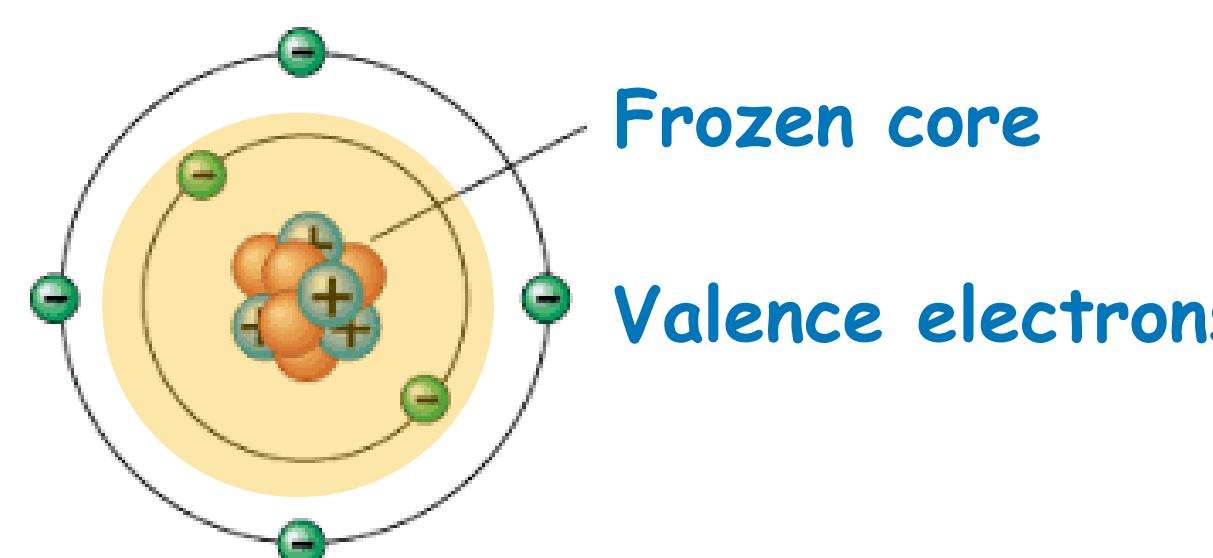
ESTESODP

$$H\Psi(r, R) = E \Psi(r, R)$$

Many-body system

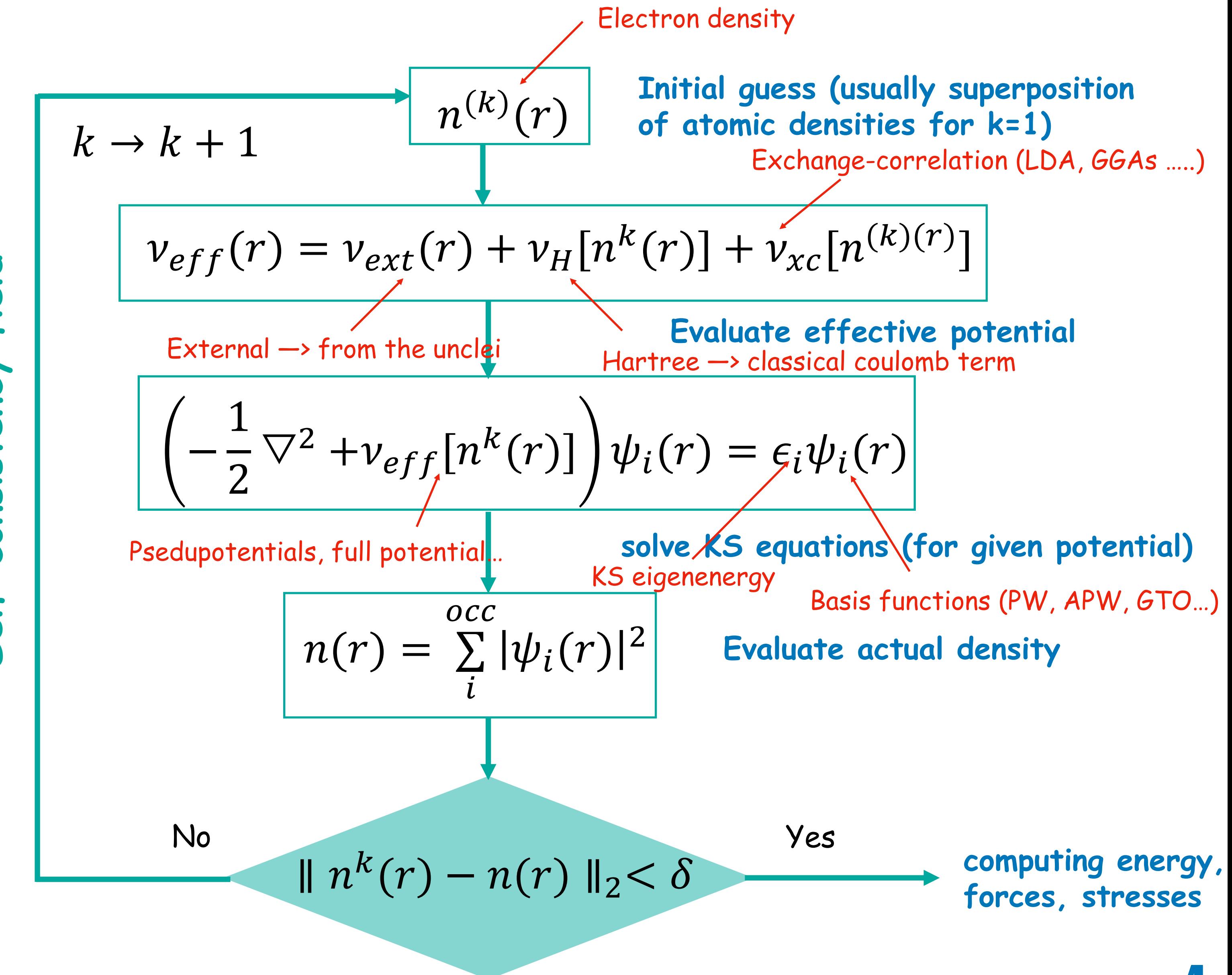


Density Function Theory (DFT)



Consider the outer electrons and freeze in the internal electrons

Self-consistency field



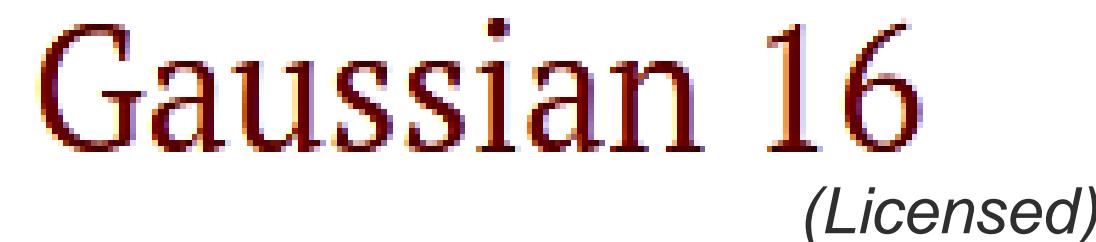
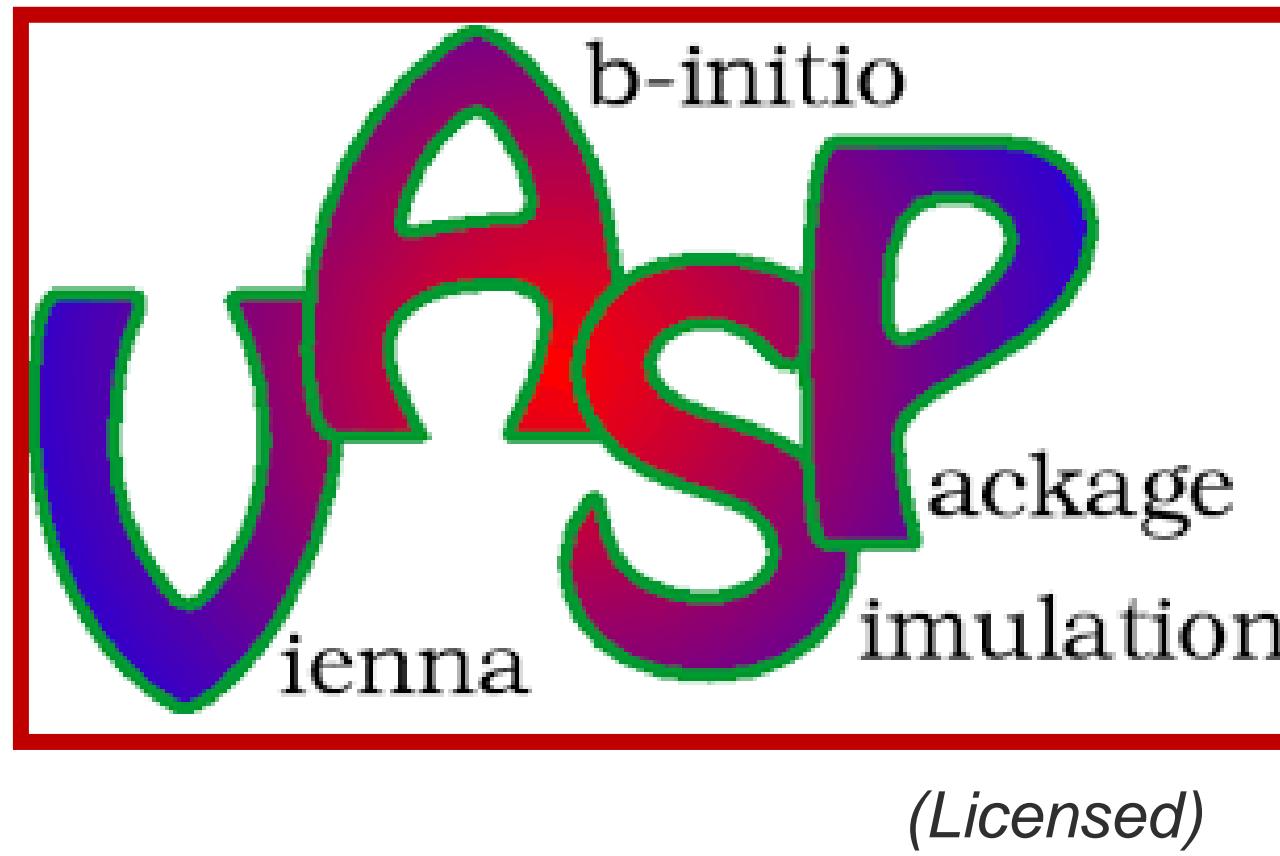
# Artisan's tools

ESTESODP



(Opensource)

- All electron potential
- LAPW
- Limited scalability



(Licensed)

- localized basis set  
(atomic orbitals)
- Molecules, all electron

- Plane wave basis
- Pseudo potential
- Highly efficient parallel computing



- Plane wave basis
- Norm-conserving & ultrasoft



(Opensource)

- All electron potential
- Norm-conserving & PAW
- Good parallel performance

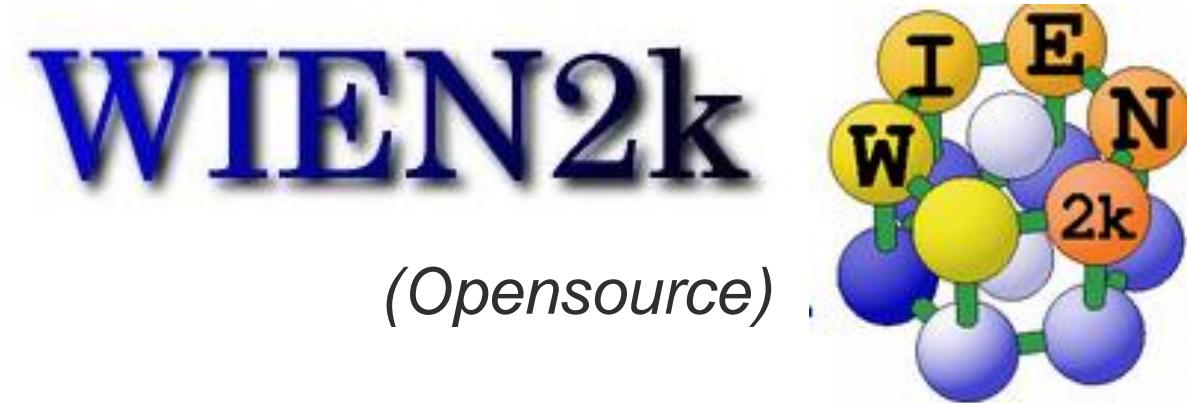
## How to choose:

1. According to the system
2. Reasonable computational cost

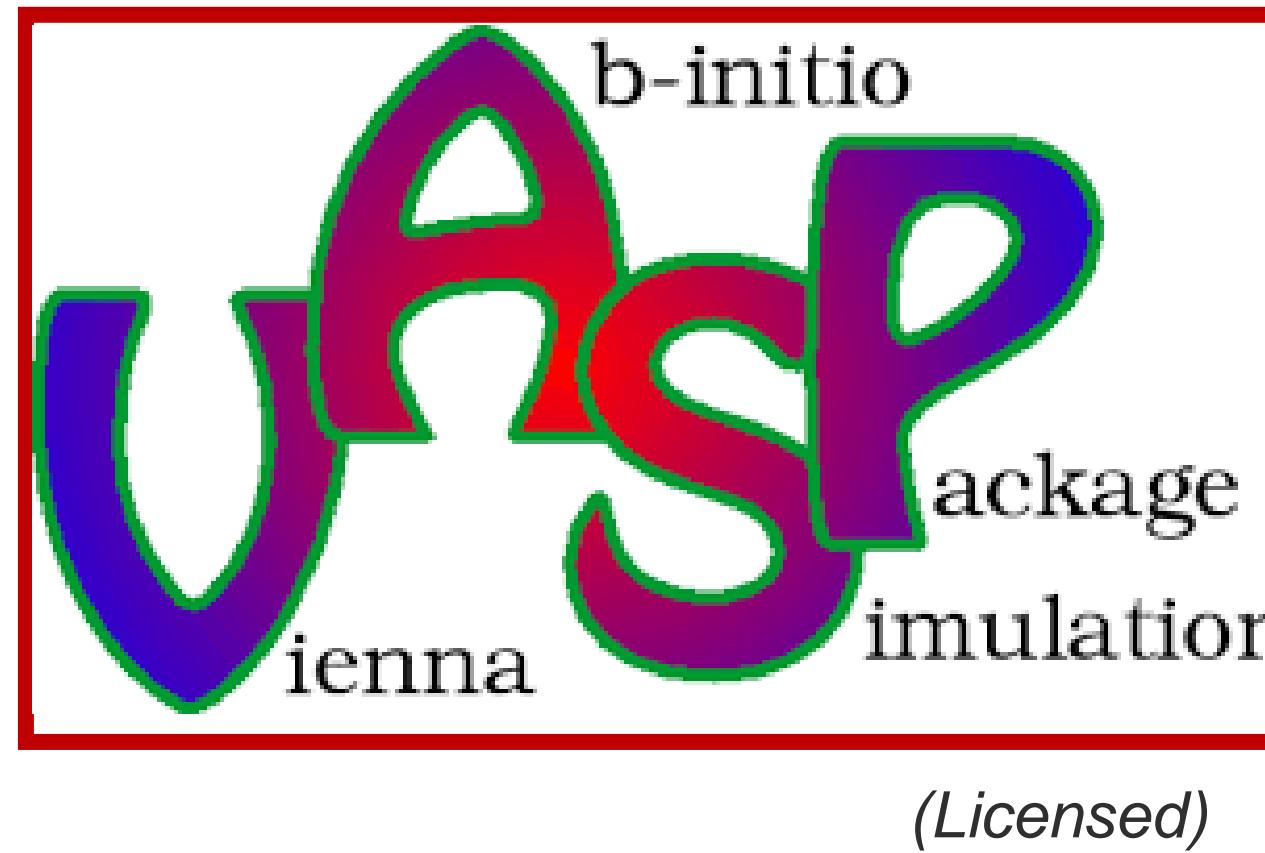
"Choice always involves a balance between accuracy & computational cost"

# Artisan's tools

ESTESODP



- All electron potential
  - LAPW
- Limited scalability



- Plane wave basis
- Pseudo potential
- Highly efficient parallel computing



- Plane wave basis
- Norm-conserving & ultrasoft



- All electron potential
- Norm-conserving & PAW
- Good parallel performance



- localized basis set
  - (atomic orbitals)
- Molecules, all electron

Files required:

Structure  
↓  
POSCAR

Potential file  
↓  
POTCAR

Control  
↓  
INCAR

grid-points  
↓  
KPOINTS

# Artisan's tools

ESTESODP

Step 1: Get the crystal structure/geometry of the compound

- Get experimental data (lattice parameter, Wycoff position)
- Get data from the existing database

Databases for structure



FIZ Karlsruhe – Leibniz Institute for Information Infrastructure



# Visualization tools

ESTESODP

https://jp-minerals.org/vesta/en/

VESTA  
Visualization for Electronic and STructural Analysis

Software > VESTA

Home About History Donate Software RIETAN-FP VESTA Change Log

1. Introduction

VESTA is a 3D visualization program for structural models, volumetric data and morphologies. Some of the novel features of VESTA are listed below.

- Deal with multiple structural models, volumetric data, and crystal models.
- Support multiple tabs corresponding to files.
- Support multiple windows with more than two tabs in the same process.

https://jmol.sourceforge.net

Jmol JSmol

Jmol: an open-source Java viewer for chemical structures in 3D with features for chemicals, crystals, materials and biomolecules

JSmol is the HTML5 modality of Jmol, able to be embedded into web pages. All the functionality of Jmol (application) is also present in JSmol.

https://www.3ds.com/products-services/biovia/products/molecular-modeling-simulation/biovia-materials-studio/

DASSAULT SYSTEMES Products Industries Learn Support Company

PRODUCTS & SERVICES BIOVIA PRODUCTS MOLECULAR MODELING & SIMULATION BIOVIA MATERIALS STUDIO

https://wiki.fysik.dtu.dk/ase/

ASE

Search docs

About Installation Getting started Tutorials Modules Command line tool Tips and tricks Gallery Release notes Contact ASE ecosystem Development Frequently Asked Questions ASE Workshop 2019

## Atomic Simulation Environment

The Atomic Simulation Environment (ASE) is a set of tools and Python modules under the [LGPL license](#).

ASE provides interfaces to different codes through [Calculators](#) which are:

```
>>> # Example: structure optimization of hydrogen molecule
>>> from ase import Atoms
>>> from ase.optimize import BFGS
>>> from ase.calculators.nwchem import NWChem
>>> from ase.io import write
>>> h2 = Atoms('H2',
...             positions=[[0, 0, 0],
...                        [0, 0, 0.7]])
>>> h2.calc = NWChem(xc='PBE')
>>> opt = BFGS(h2)
>>> opt.run(fmax=0.02)
BFGS:  0 19:10:49 -31.435229  2.2691
BFGS:  1 19:10:50 -31.490773  0.3740
BFGS:  2 19:10:50 -31.492791  0.0630
BFGS:  3 19:10:51 -31.492848  0.0023
>>> write('H2.xyz', h2)
>>> h2.get_potential_energy()
-31.492847800329216
```

### Supported calculators

ab initio asap Atomistica Big DFT exciting EMT FHI-aims Fleur GAMESS GROMACS Hb KIM LAMMPS NWChem QMCPACK Q-CHEM QUANTUMESPRESSO siesta TURBOMOLE VASP xTB ORCA ACE-Molecule am DMol3 Gaussian Grimme DFT-D3gulp Mopac qmmm tip3p ~deMon-Nano

BIOVIA MATERIALS STUDIO  
AN INTEGRATED, MULTI-SCALE MODELING ENVIRONMENT

Modeling & Simulation for Next-Generation Materials

BIOVIA Materials Studio is a complete modeling and simulation environment designed to allow researchers in materials science and chemistry to predict and understand the relationships of a material's atomic and molecular structure with its properties and behavior. Using Materials Studio, researchers in many industries are engineering better performing materials of all types, including catalysts, polymers, composites, metals, alloys, pharmaceuticals, batteries and more.

Advancing Materials Development through the use of Materials Studio

www.xcrysden.org

XcrySDen ... X-window CRYstalline Structures and DENsities

Home | About | Description | Documentation | Download | News | Links

**XCrySDen**

XCrySDen is a crystalline and molecular structure visualisation program aiming at display of isosurfaces and contours, which can be superimposed on crystalline structures and interactively rotated and manipulated. It runs on GNU/Linux.

XCrySDen has been also ported to Mac OS (requires X11) and Windows (requires either CYGWIN or WSL).

The name of the program stands for [Crystalline Structures and Densities](#) and [X](#) because it runs under the X-Window environment.

[Read more...](#) | [See screenshots ...](#)

Latest version: [1.6.2](#)

**XCrySDen mailing list**

XCrySDen mailing list is an open mailing list where XCrySDen related issues can be discussed among users.

[Subscribe](#) | [Archives](#)

**News**

[28 Oct 2019] [XCrySDen-1.6.2 released](#).

[Read more...](#)

**Terms of use**

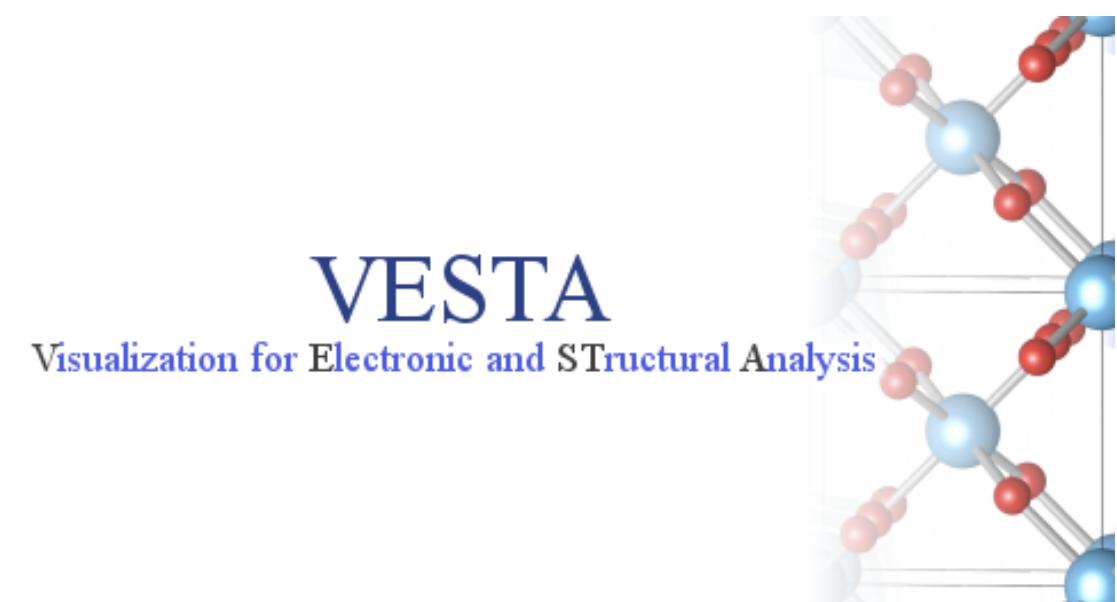
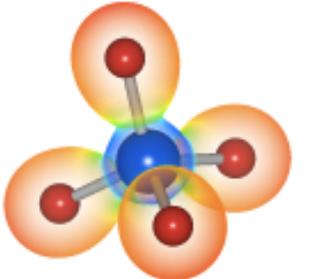
XCrySDen is released under the GNU General Public License.

Whenever graphics generated by XCrySDen are used in scientific publications, it shall be greatly appreciated to include an explicit reference. The preferred form is the following:

[ref] A. Kokalj, *J. Mol. Graphics Modelling*, **1999**, *17*, 176–179. Code available from <http://www.xcrysden.org/>.

# Creating a structure

ESTESODP



Download the FREE software VESTA using URL:

<http://jp-minerals.org/vesta/en/download.html>

- Windows
- Linux
- Mac OS

1. File -> New structure
2. Open the “Unit Cell” tab
3. Type in the crystal system, space group and lattice parameters of your compound (i.e., Mn<sub>3</sub>Ga).
4. In the “Structure Parameters” tab add labels and atom positions of the atoms in your unit cell.
5. At this stage you should save what you have done as .vesta and possibly also export to POSCAR (assuming that you run calculations with VASP).
6. Save image in different formats using, File >> Export Raster image.

# Basic structure

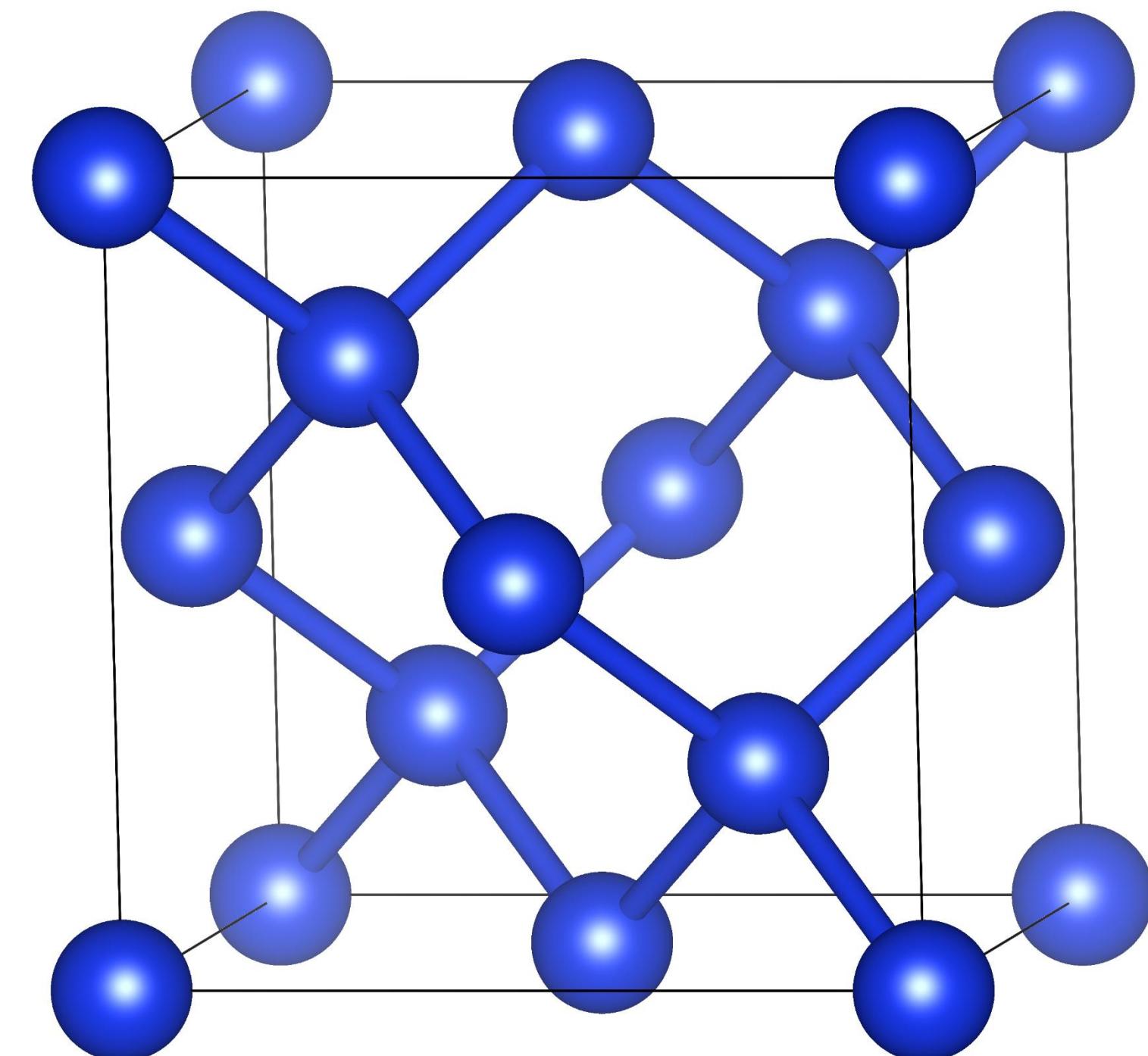
ESTESODP

Conventional cell:

Crystal structure: Diamond cubic

Space Group: Fd-3m (227)

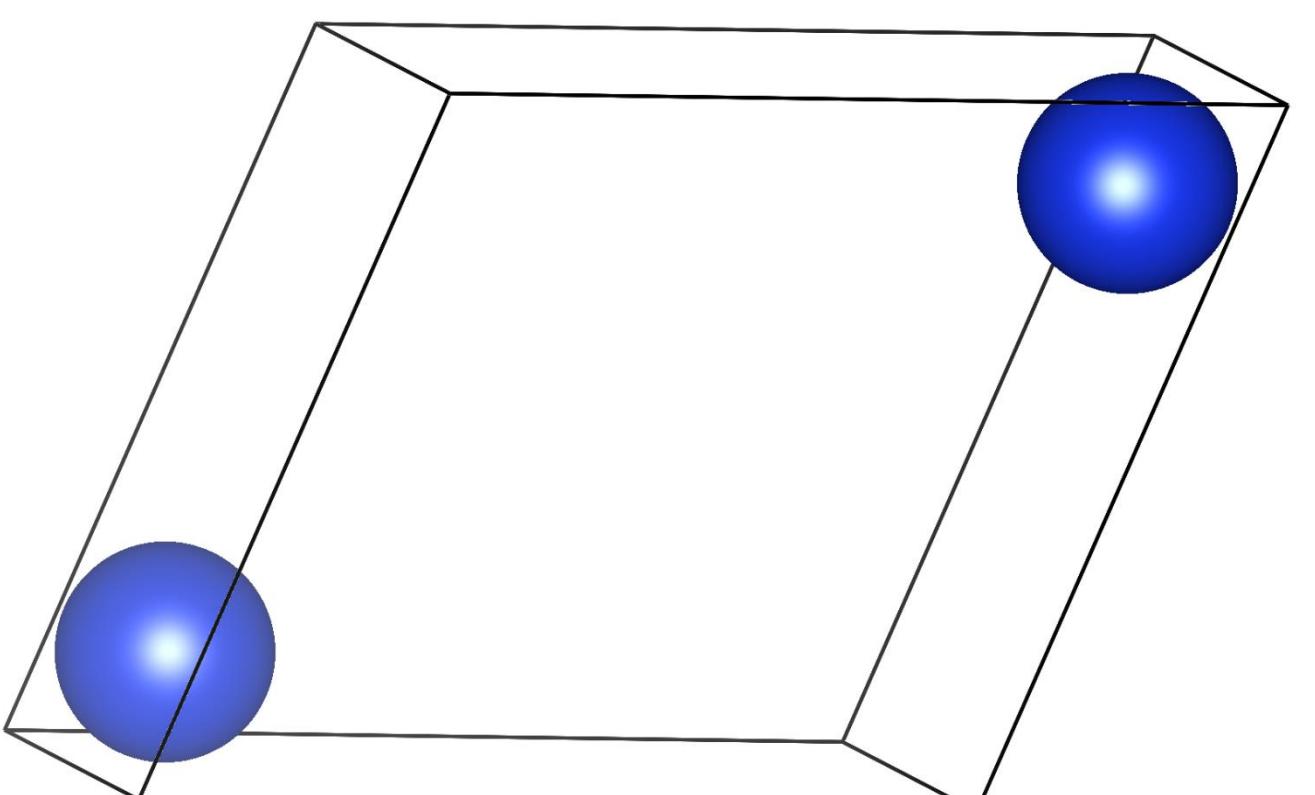
$a = b = c = 5.44 \text{ \AA}$ ,  $\alpha = \beta = \gamma = 90^\circ$



Primitive Cell:

Total Atoms: 2

$a = b = c = 3.849$ ,  $\alpha = \beta = \gamma = 60.000^\circ$



Commercial

Pearson's Crystal Data

Materials Studio

<https://materialsproject.org/>

<http://aflowlib.org/>

free

# Example

Linux server  
↓  
**Server IP:** 10.111.1.12  
**User Name:** estesodp.g1  
**password:** tut\_estesodp\_2024

Use secure shell to connect  
**ssh estesodp.g1@10.111.1.12**

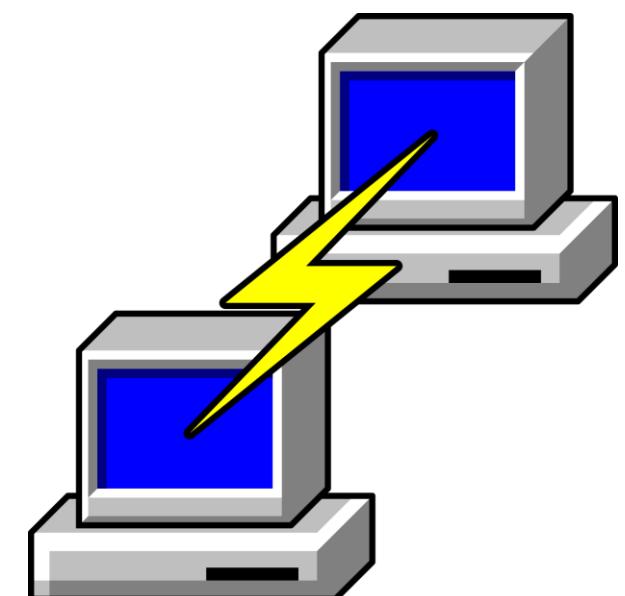
or

**ssh -oHostKeyAlgorithms=+ssh-dss estesodp.g1@10.111.1.12**

For windows - use putty

```
krishnarajmaya --zsh -- 80x24
Last login: Tue Sep 26 13:49:19 on ttys000
(base) [REDACTED] ~ ssh estesodp@10.111.1.12

• PC login password: student
• WiFi details:
  U: estesodp.guest1
  .
  p: Hks3U$M
```



# Example

Create a new folder with your name: **mkdir <your\_name>**

Move to the new folder: **cd <your\_name>**

Create a folder "Example\_01": **mkdir Example\_01**

Open a new file for editing using  
"vim" editor: **vi POSCAR**

**cp ~/input\_files/POSCAR path\_to\_your\_folder/**

Press I to start inserting text

To know that path to your current directory do "pwd"

Press "escape" and then ":wq" to save and exit

Press "escape" and then ":q!" to exit without saving

The screenshot shows a terminal window with a black background and white text. At the top, there's a title bar with a red dot, a yellow dot, a green dot, and a blue dot, followed by the text "/example\_01/Si – ssh MM74...". Below the title bar, the text "System Si" and "5.430" is visible. The main area of the terminal shows the content of a POSCAR file:

```
0.5 0.5 0.0
0.0 0.5 0.5
0.5 0.0 0.5

Si
2
cart
0.00 0.00 0.00
0.25 0.25 0.25

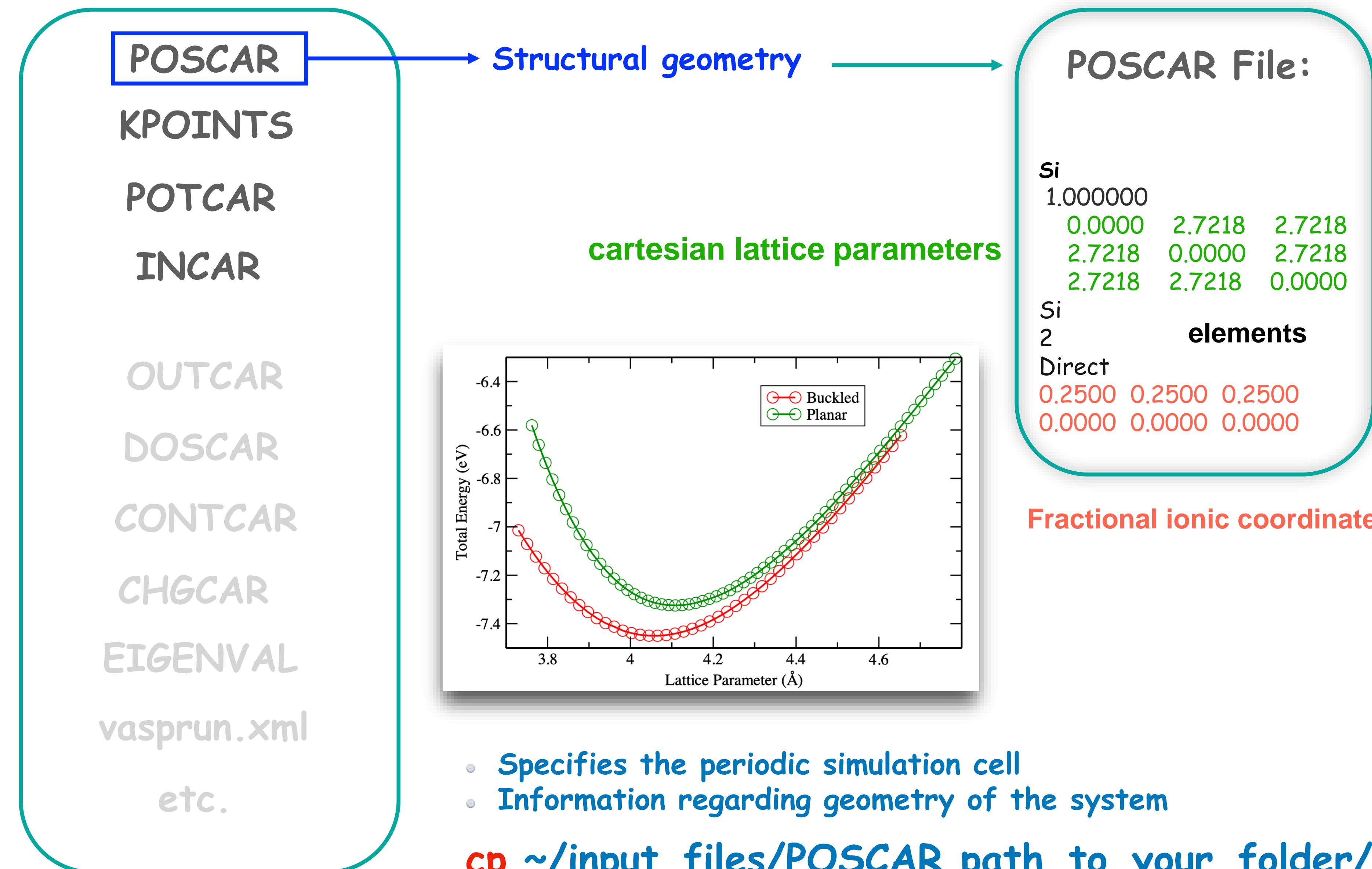
~
~
~

-- INSERT --
```

At the bottom right of the terminal window, there are status indicators: "1,10" and "A11".

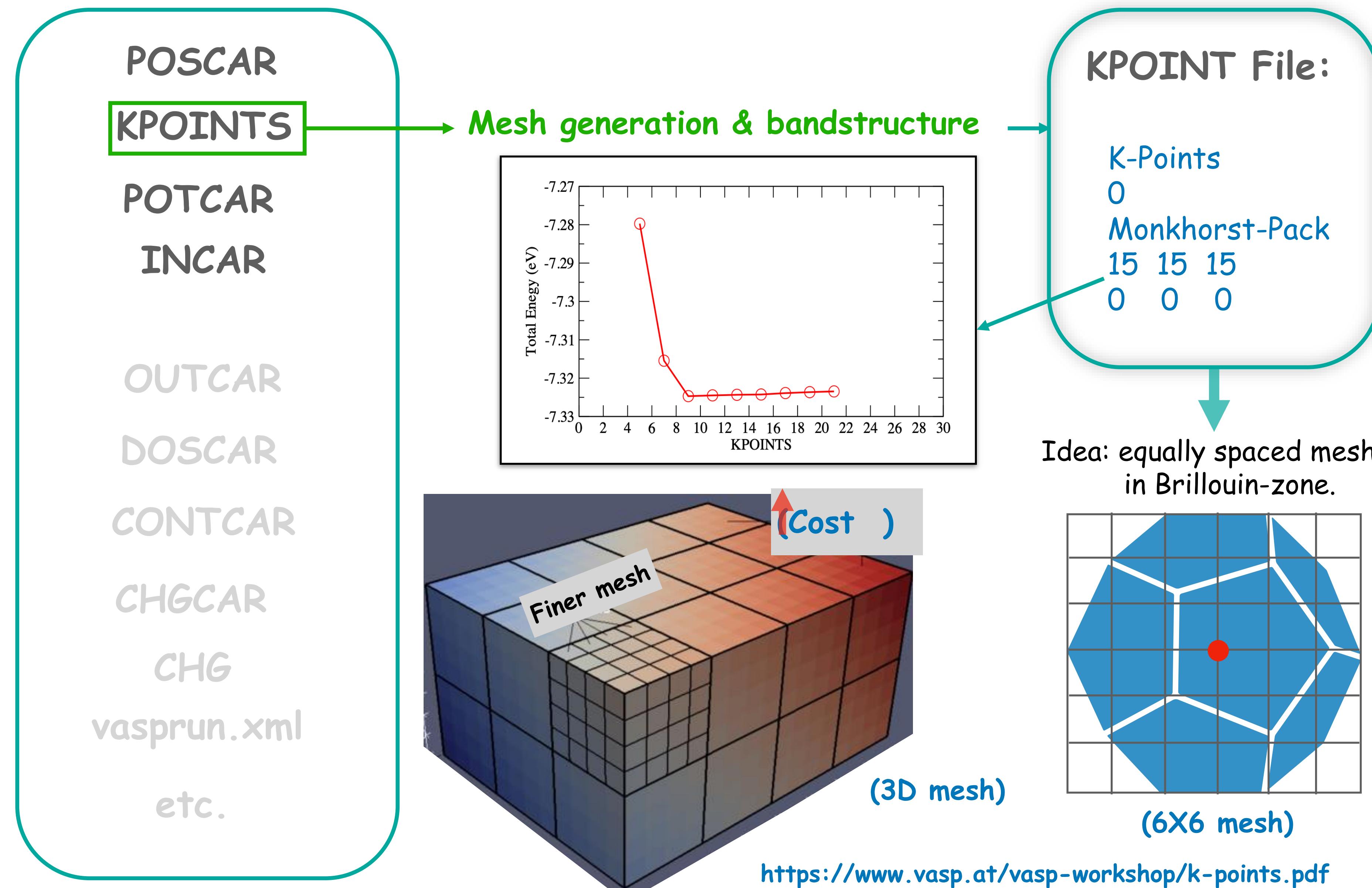
# VASP Input files

ESTESODP



# VASP Input files

ESTESODP



# Reciprocal lattice and Brillouin zones (KPOINTS)

ESTESODP

The screenshot shows the MaterialsCloud SeeK-path tool interface. The top navigation bar includes links for LEARN, WORK, DISCOVER, EXPLORE, and ARCHIVE. The main title is "SeeK-path: the k-path finder and visualizer". The interface is divided into several sections:

- Brillouin zone (go to coordinates):** A 3D visualization of the Brillouin zone with axes  $b_1$ ,  $b_2$ , and  $b_3$ . It features points labeled G, R, S, N, T, and M.
- Primitive structure (go to coordinates):** A 3D visualization of the primitive structure.
- Reciprocal space and Brillouin-zone information:** Displays reciprocal cell vectors ( $1/\text{\AA}$ )

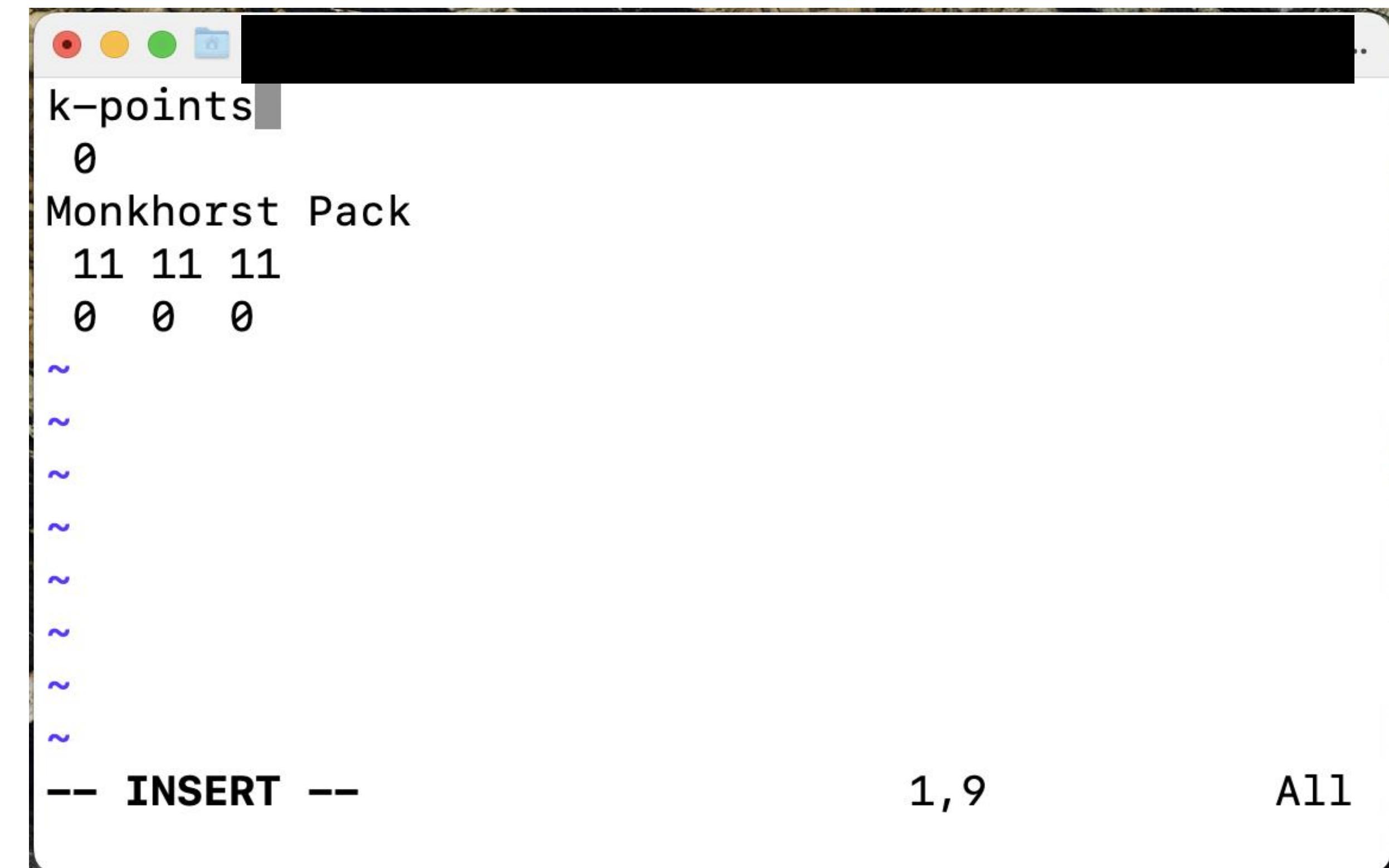
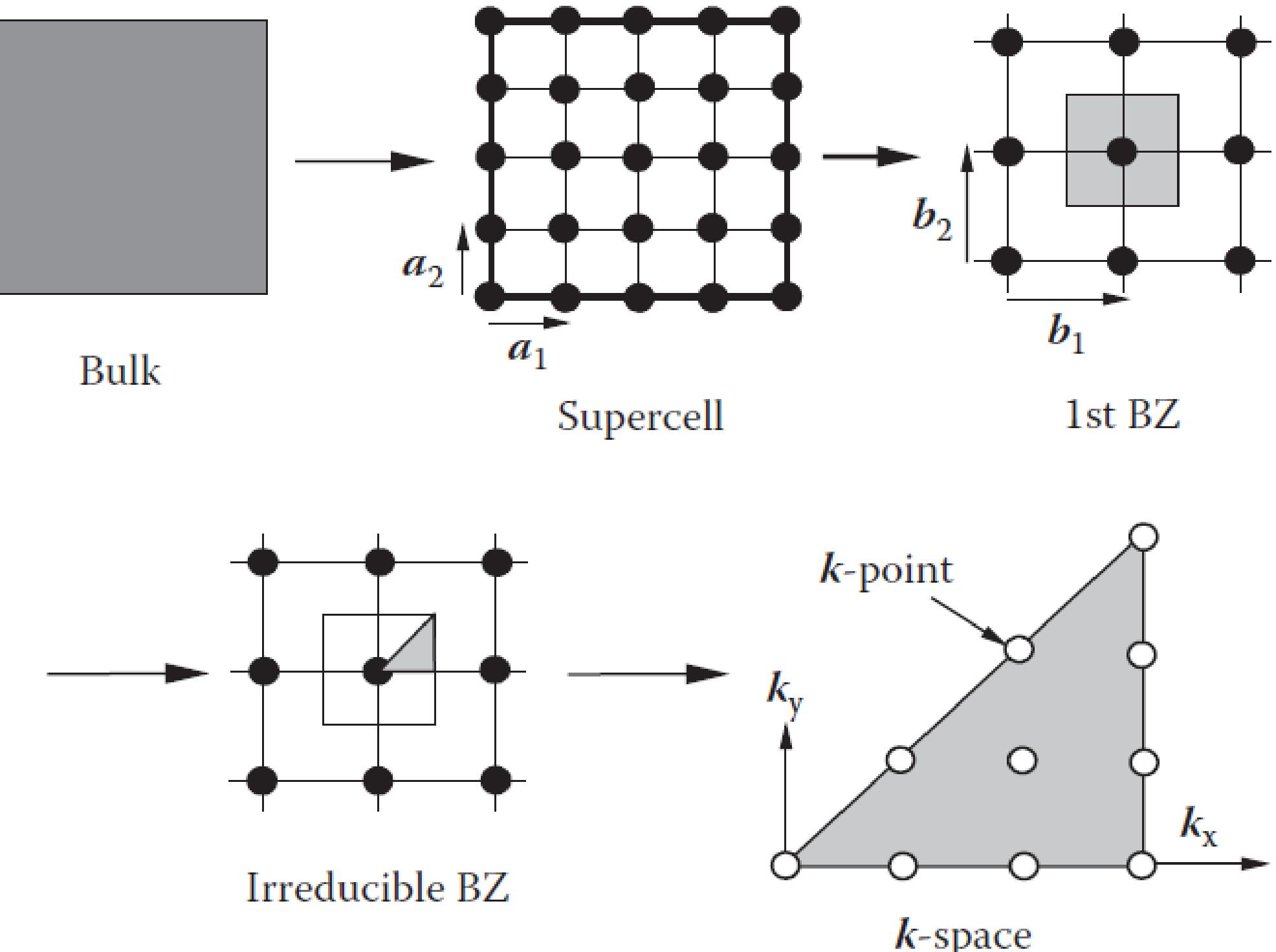
<b>b</b>	<b>x</b>	<b>y</b>	<b>z</b>
$b_1$	-0.0000000000	1.6659875218	0.9131180379
$b_2$	1.6659875218	0.0000000000	0.9131180379
- Structure information (primitive cell):** Displays crystal structure information, Bravais lattice type (tl), extended Bravais lattice symbol (tl2 with inversion symmetry), and spacegroup (14/mmm number 139).

On the left, there are sections for "What SeeK-path does", "SeeK-path definitions and advantages", "Upload your structure", "Otherwise, pick an example", "How to cite", and "Note: if you want to use the code on your computer, you can repository".

Or generate KPOINTS using VASPKIT

# K-grid choices

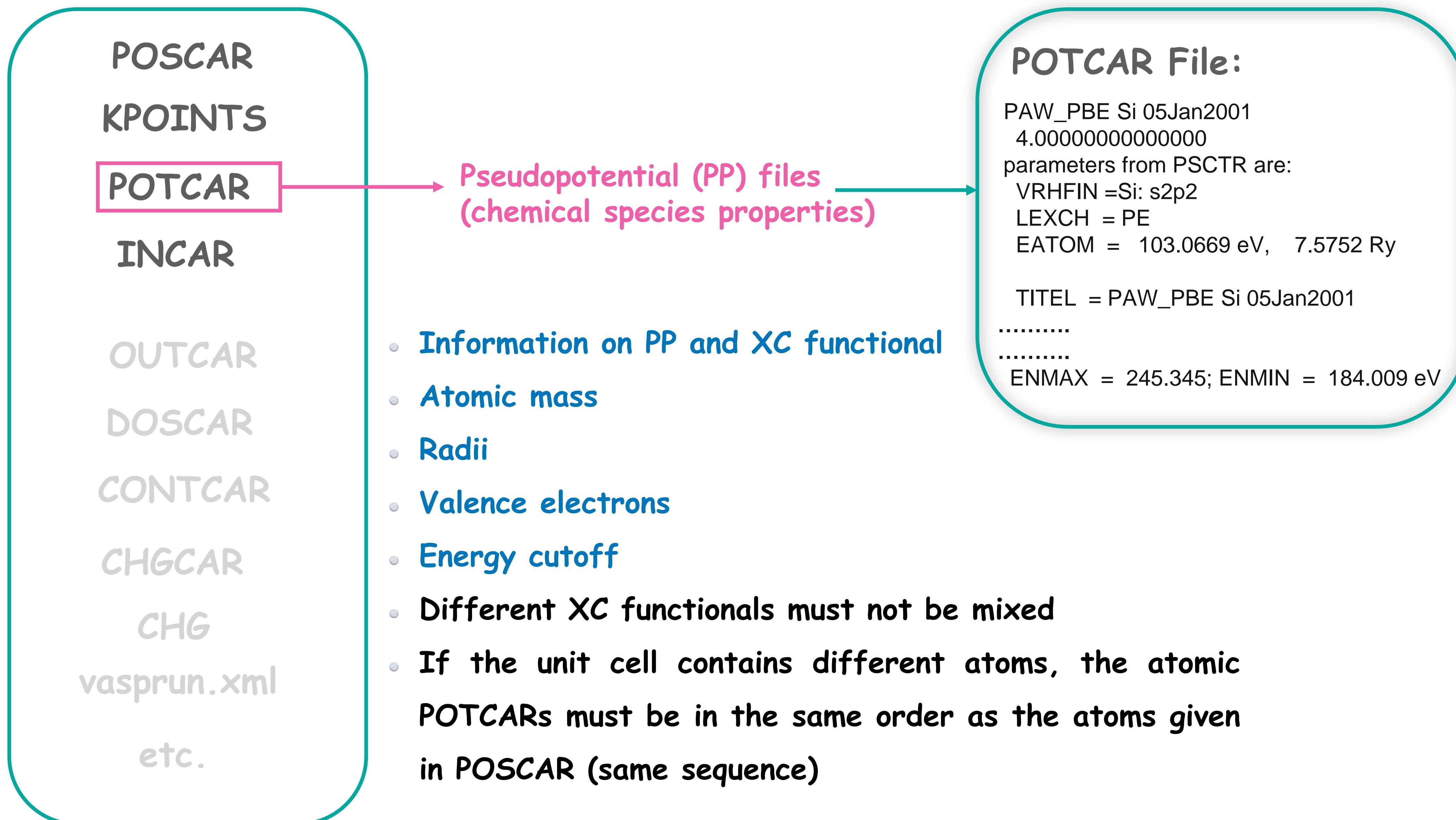
ESTESODP



```
cp ~/input_files/KPOINTS path_to_your_folder/
cp ~/input_files/KPOINTS ./
```

# VASP Input files

ESTESODP



# Example

Copy the required potentials to POTCAR

```
cat <path_to_potentials_folder>/Si/POTCAR >> POTCAR  
cat ~/input_files/Tut-1/Si/POTCAR >> POTCAR  
cp ~/input_files/Tut-1/Si/POTCAR ./
```

Or

use vasakit to generate POTCAR

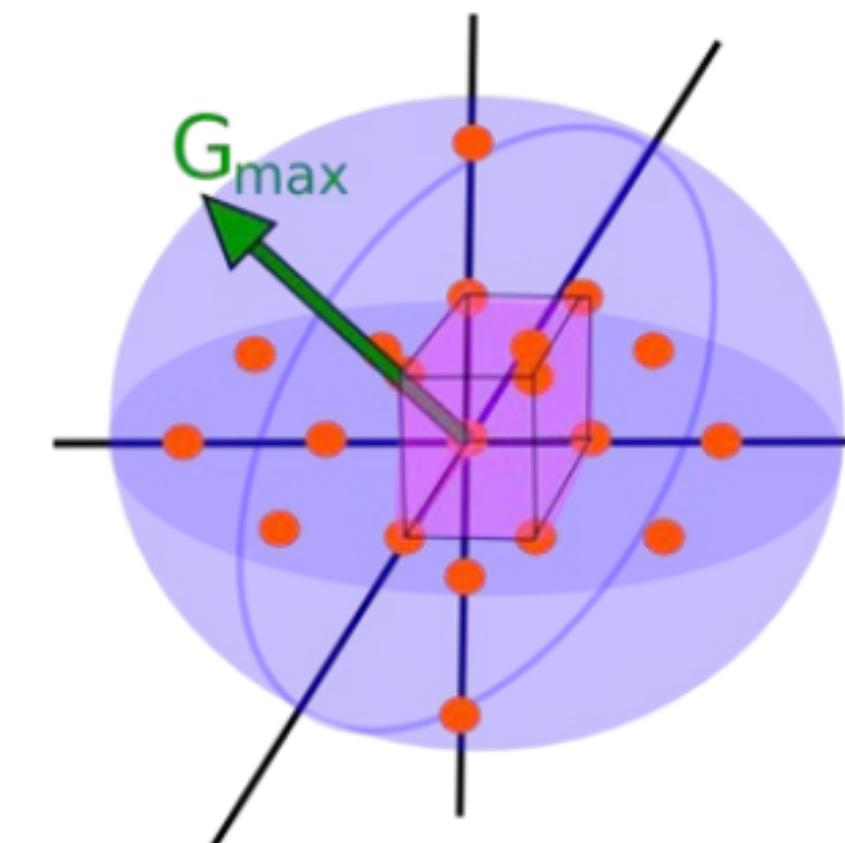
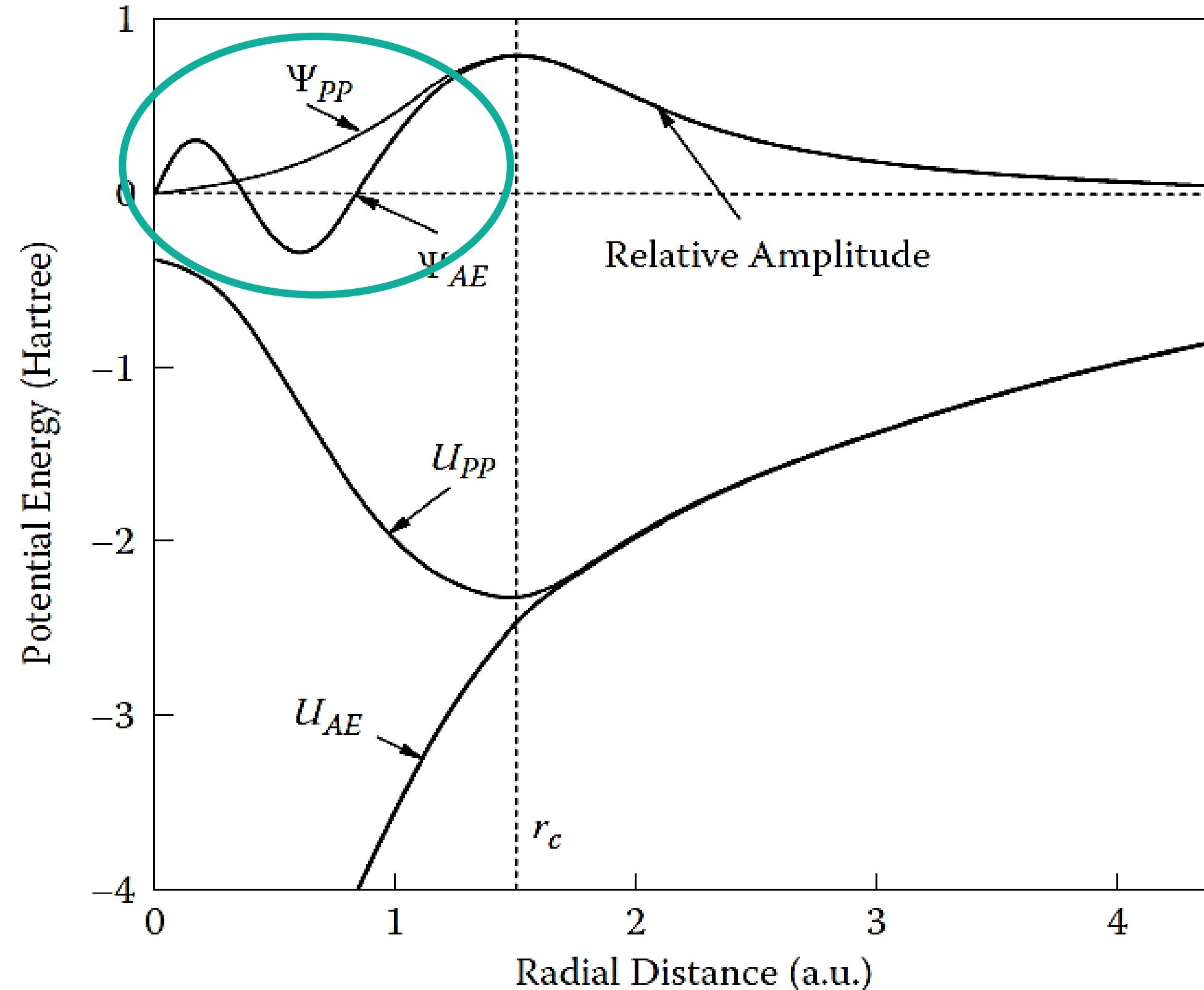
- \* Potentials for all the elements present in the POSCAR should be copied to a single POTCAR in the destination folder with same order as given in POSCAR

```
PAW_PBE Si 05Jan2001  
4.0000000000000000  
parameters from PSCTR are:  
VRHFIN =Si: s2p2  
LEXCH = PE  
EATOM = 103.0669 eV, 7.5752 Ry  
  
TITEL = PAW_PBE Si 05Jan2001  
LULTRA = F use ultrasoft PP ?  
IUNSCR = 1 unscreen: 0-lin 1-nonlin 2-  
no  
RPACOR = 1.500 partial core radius  
@  
-- INSERT -- 1, 24 Top
```

# Pseudo-potentials and energy cutoffs (POTCAR)

ESTESODP

Pseudo-potentials reduces the computational cost while maintaining the accuracy



$$E_{cutoff} = \frac{G_{max}^2}{2}$$

Plane waves within the Kinetic energy cutoff are considered for calculation

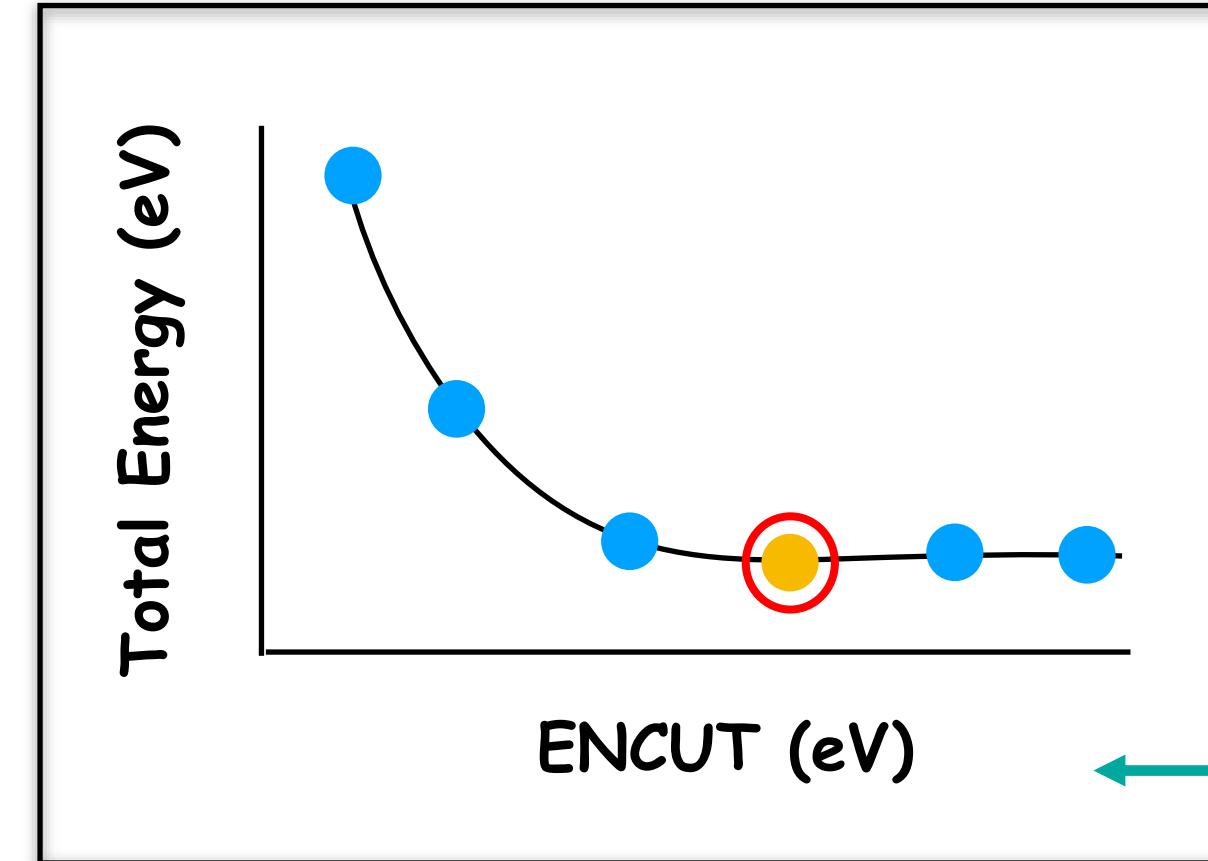
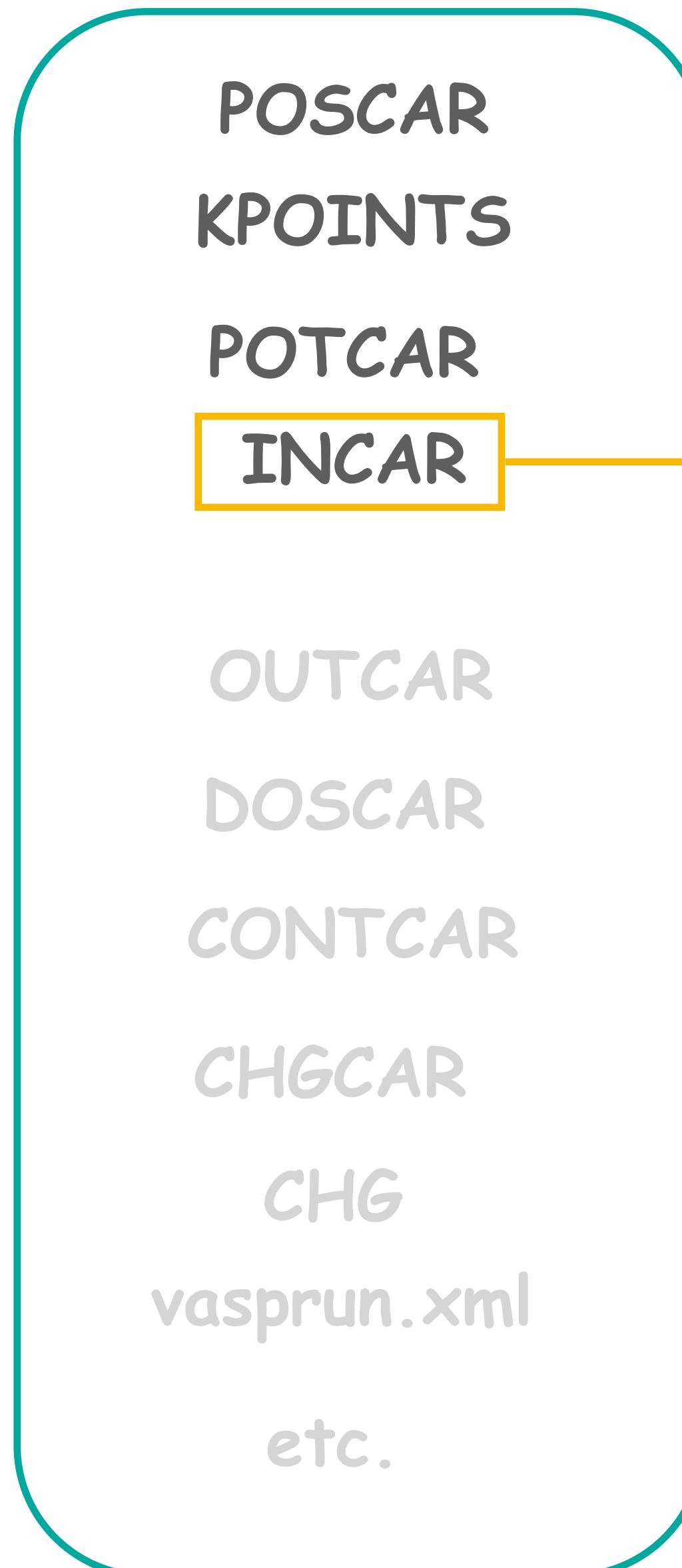
PAW\_PBE Si 05Jan2001  
4.000000000000000  
parameters from PSCTR are:  
VRHFIN =Si: s2p2  
LEXCH = PE  
EATOM = 103.0669 eV, 7.5752 Ry  
  
TITEL = PAW\_PBE Si 05Jan2001  
LULTRA = F use ultrasoft PP ?  
IUNSCR = 1 unscreen: 0-lin 1-nonlin 2-no  
RPACOR = 1.500 partial core radius  
POMASS = 28.085; ZVAL = 4.000 mass and valenz  
RCORE = 1.900 outmost cutoff radius  
RWIGGS = 2.480; RWIGS = 1.312 wigner-seitz radius (au A)  
ENMAX = 245.345; ENMIN = 184.009 eV  
ICORE = 2 local potential  
LCOR = T correct aug charges  
LPAW = T paw PP

VASP provides PBE - PAW Pseudo-potentials

$ENCUT \approx 1.3 \times ENMAX$

# VASP Input files

ESTESODP



Central file

## INCAR File:

- `ISTART` → scf/non-scf
- `ENCUT` → energy cutoff
- `NSW` → ionic steps
- `ISIF` → degree of freedom
- `EDIFF` → break conditions
- `EDIFFG` → break conditions
- etc.

- The default values set by VASP itself are a good choice to do standard calculations.
- Tags can be modified according to your properties calculations (scf calculations, DOS, band structure, etc.)

INCAR tags usage found in vaspWiki:

[https://www.vasp.at/wiki/index.php/Category:INCAR\\_tag](https://www.vasp.at/wiki/index.php/Category:INCAR_tag)

# INCAR tags

ESTESODP

**ISTART** = 0 | 1 | 2 | 3 To optimize the wave functions or not

Default: **ISTART** = 1 if a WAVECAR file exists  
= 0 else

Description: **ISTART** determines whether or not to read the WAVECAR file.

**ICHARG** = 0 | 1 | 2 | 4 To optimize the Charge densities or not

Default: **ICHARG** = 2 if **ISTART**=0  
= 0 else

Description: **ICHARG** determines how VASP constructs the *initial* charge density.

**ALGO** = Normal | VeryFast | Fast |

Default: **ALGO** = Normal

Description: the **ALGO** tag is a convenient option to specify the electronic minimization algorithm

**PREC** = Low | Medium | High | Normal | Single | Accurate

Default: **PREC** = Medium for VASP.4.X  
= Normal for VASP.5.X

Description: **PREC** specifies the "precision"-mode.

<b>ISIF</b>	calculate		degrees-of-freedom		
	forces	Stress tensor	positions	cell shape	cell volume
0	yes	no	yes	no	no
1	yes	trace only	yes	no	no
2	yes	yes	yes	no	no
3	yes	yes	yes	yes	yes
4	yes	yes	yes	yes	no

**SIGMA** = [real]

Default: **SIGMA** = 0.2

Description: **SIGMA** specifies the width of the smearing in eV.

**EDIFF** = [real]

Default: **EDIFF** =  $10^{-4}$

Description: **EDIFF** specifies the global break condition for the electronic SC-loop.

# A basic example of INCAR

MM 747

```
MM747_2023@dendrite:~/example_01/Si - ssh MM74...  
ISTART=0 # starting from scratch  
ISMEAR=0 # partial occupancies  
SIGMA=0.1 # metal 0.1– 0.3  
ENMAX=270 # Energy maximum of wave function  
ISPIN=2 # Spins will accounted  
IBRION=2 # Relax the atoms  
NSW=100 # Number of iterations  
ISIF=3 # Atomic coordinate optimization  
EDIFF=0.1E-03 # Precision of optimization  
EDIFFG=-0.001 # Precision of optimization  
LORBIT=11 # Writing wavefunctions  
  
~  
"INCAR" 12L, 383C 1,1 All
```

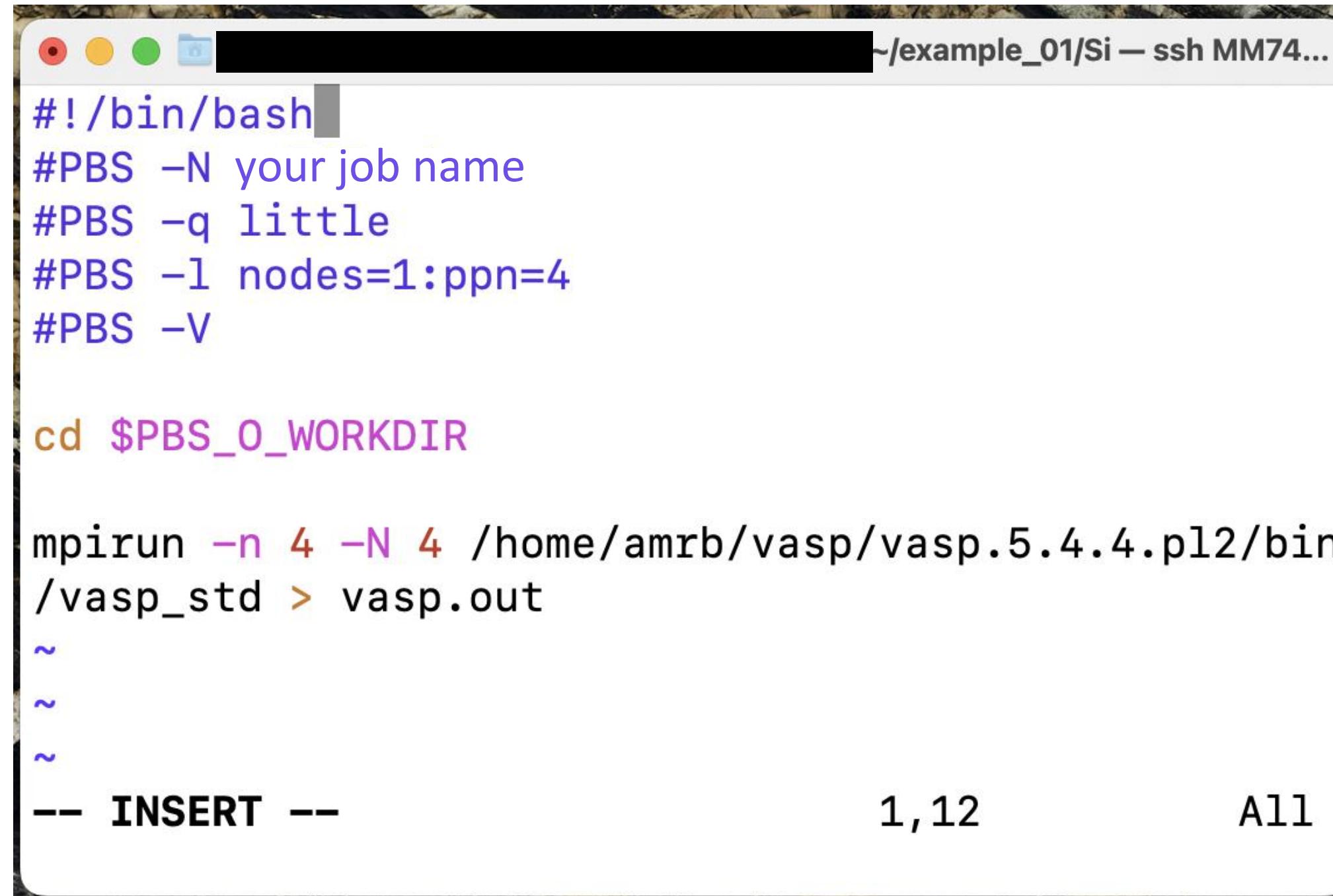
ISPIN = 1 for non-spin-polarized calculation  
MAGMOM (will be discussed Tutorial-3)

**cp** ~/input\_files/Tut-1/Si/INCAR path\_to\_your\_directory  
**cp** ~/input\_files/Tut-1/Si/INCAR ./

# Run a calculation

ESTESODP

To execute a vast calculation in parallel environment with multiple processors, a job handling system is used in cluster environments



The screenshot shows a terminal window with a black background and white text. The title bar reads '~ /example\_01/Si – ssh MM74...'. The script content is as follows:

```
#!/bin/bash
#PBS -N your job name
#PBS -q little
#PBS -l nodes=1:ppn=4
#PBS -V

cd $PBS_O_WORKDIR

mpirun -n 4 -N 4 /home/amrb/vasp/vasp.5.4.4.pl2/bin
/vasp_std > vasp.out
~
~
~

-- INSERT --           1,12          All
```

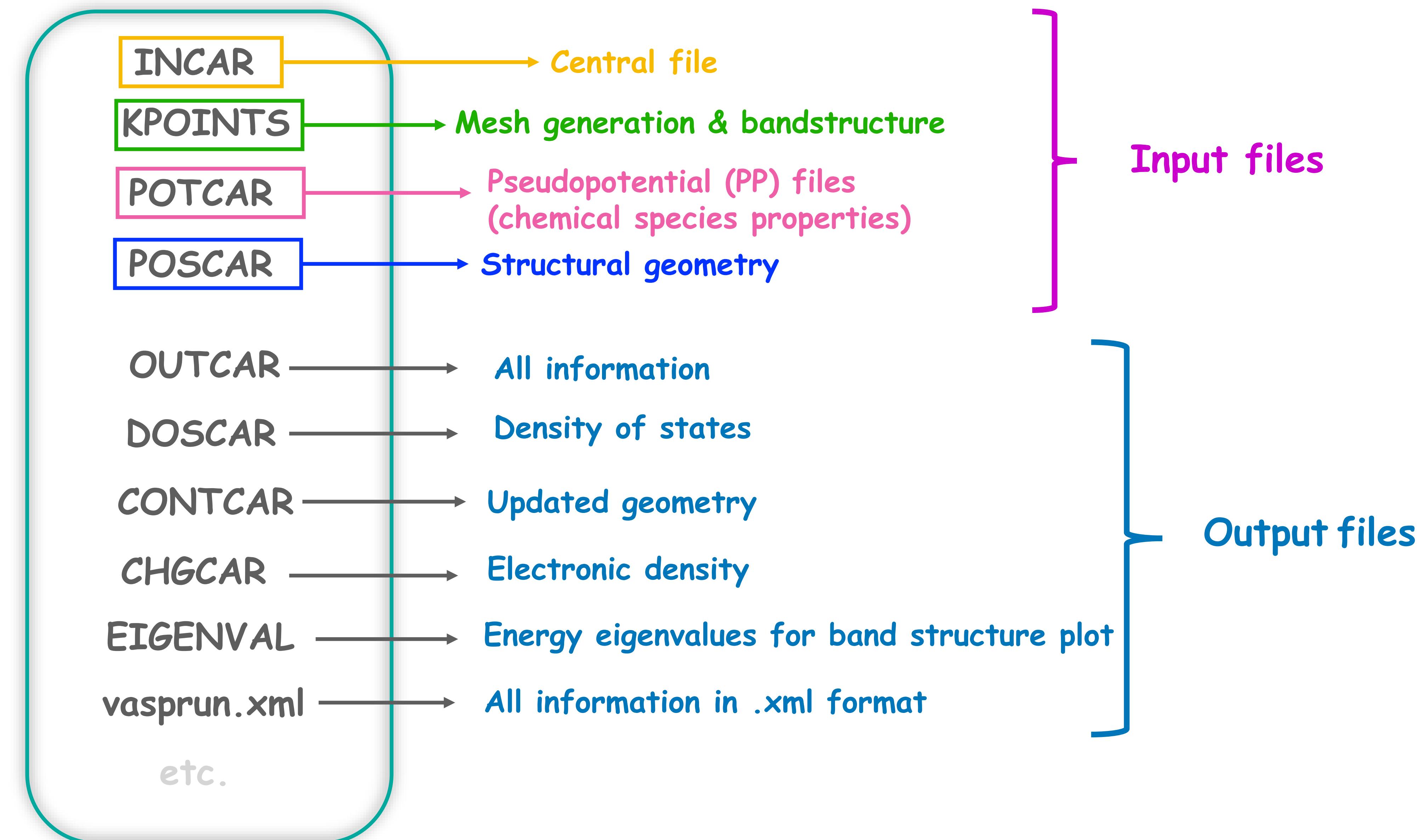
**cp ~ /input\_files/Tut-1/Si/job.sh path\_to\_your\_directory**  
**cp ~ /input\_files/Tut-1/Si/ job.sh ./**

Submit the job: **qsub job.sh**

Check job status: **qstat**

# VASP Input-Output files

ESTESODP



# Contents in Output files

ESTESODP

**CONTCAR**

Relaxed structural parameters  
Lattice vectors  
Atom coordinates

**CHGCAR**

Charge densities at each k - point  
Occupancies

**WAVECAR**

Wavefunctions  
Fermi weights  
No of bands

**EIGENVAL**

Eigen values at each k - point

**IBZKPT**

k - points used in the  
calculation based on  
symmetry of the structure

**DOSCAR**

Density of states and  
Integrated density of  
states at each k - point

**OUTCAR**

Vasprun.xml

Total energies  
Forces on each atom  
KS - Eigen values  
Stress tensors  
Dielectric constants  
Magnetic moments  
Minimization steps

**vasp.out**

Errors, warnings and  
summary of calculation  
after each iteration

# To check convergence and get total energy

```
=====
Total CPU time used (sec): 122.821
```

```
        User time (sec): 120.971
```

```
        System time (sec): 1.851
```

```
Elapsed time (sec): 129.229
```

```
Maximum memory used (kb): 69744.
```

```
Average memory used (kb): N/A
```

```
Minor page faults: 11780
```

```
Major page faults: 6
```

```
Voluntary context switches: 154463
```

To get the total DFT energy

```
Command: grep TOTEN OUTCAR|tail -1
```

```
free energy TOTEN = -10.84939200 eV
```

## POSITION

```
3.90691 2.76260 6.76696
```

```
0.55813 0.39466 0.96671
```

## TOTAL-FORCE (eV/Angst)

```
0.000000 -0.000000 0.000000
```

```
-0.000000 0.000000 -0.000000
```

```
total drift:
```

```
-0.000000 0.000000 -0.000000
```

# ENCUT Convergence

ESTESODP

Create INCAR\_0 with a place holder

```
ISTART = 0
ISMEAR = 0
SIGMA = 0.1
ENCUT = <e>
ISPIN = 2
IBRION = 2
NSW = 100
ISIF = 2
LWAVE = .FALSE.
LCHARG = .FALSE.
EDIFF = 0.1E-03
EDIFFG = -0.001
```

Modify job script to run in a loop

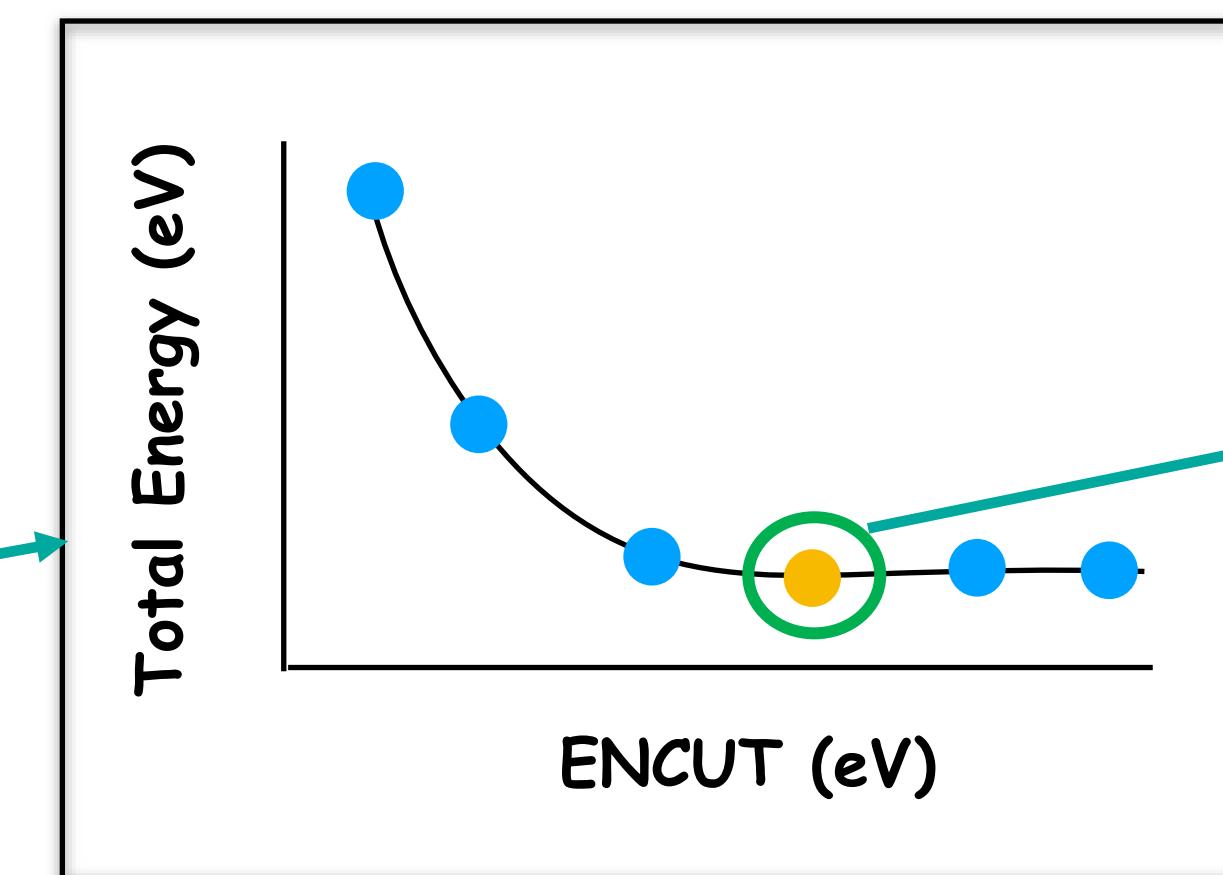
```
#!/bin/bash
#PBS -N MM747_ec
#PBS -q little
#PBS -l nodes=1:ppn=4
#PBS -V

cd $PBS_O_WORKDIR

for e in {200..600..50}; do
cat INCAR_0 | sed "s/<e>/$e/g" > INCAR
mkdir $e
cp INCAR KPOINTS POSCAR POTCAR $e/.
cd ./$e
mpirun -n 4 -N 4 /home/amrb/vasp/vasp.6.2.0/bin/vasp_std > vasp.out
cd ..
done
```

Submit the job: **qsub job.qsub**

Once the job is completed, Plot Total energies with respect to ENCUT values

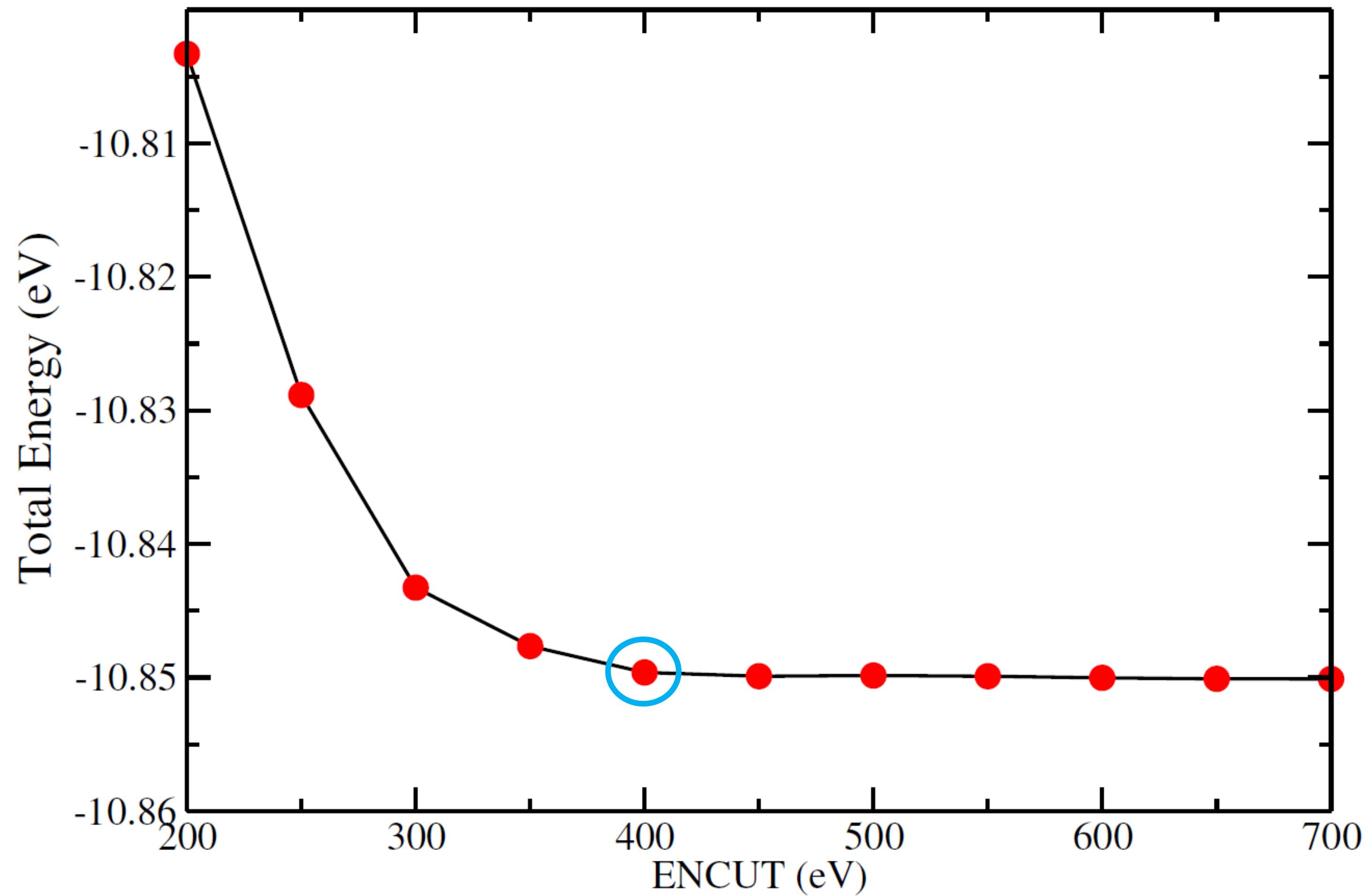


Chosen ENCUT value for further calculations

Computational cost  $\propto$  ENCUT

# ENCUT convergence for Si diamond cubic

ESTESODP



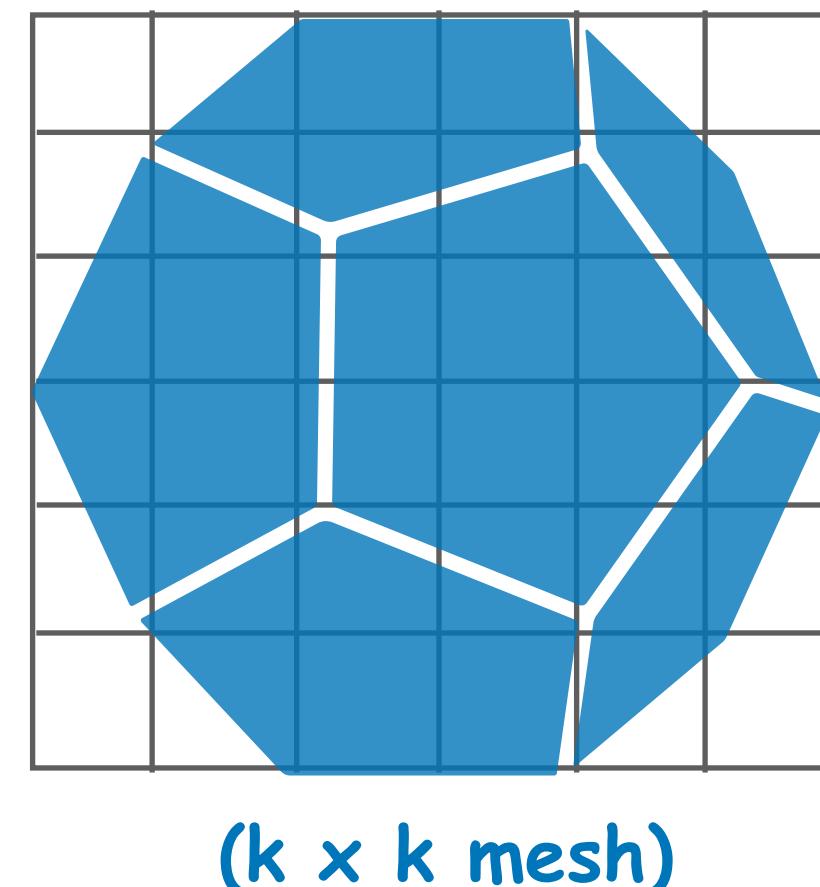
# KPOINTS Convergence

ESTESODP

Create KPOINTS\_0 with a place holder

```
K-Points  
0  
Monkhorst-Pack  
<k> <k> <k>  
0 0 0
```

Idea: equally spaced mesh  
in Brillouin-zone.



INCAR same as  
previous with  
converged ENCUT  
value

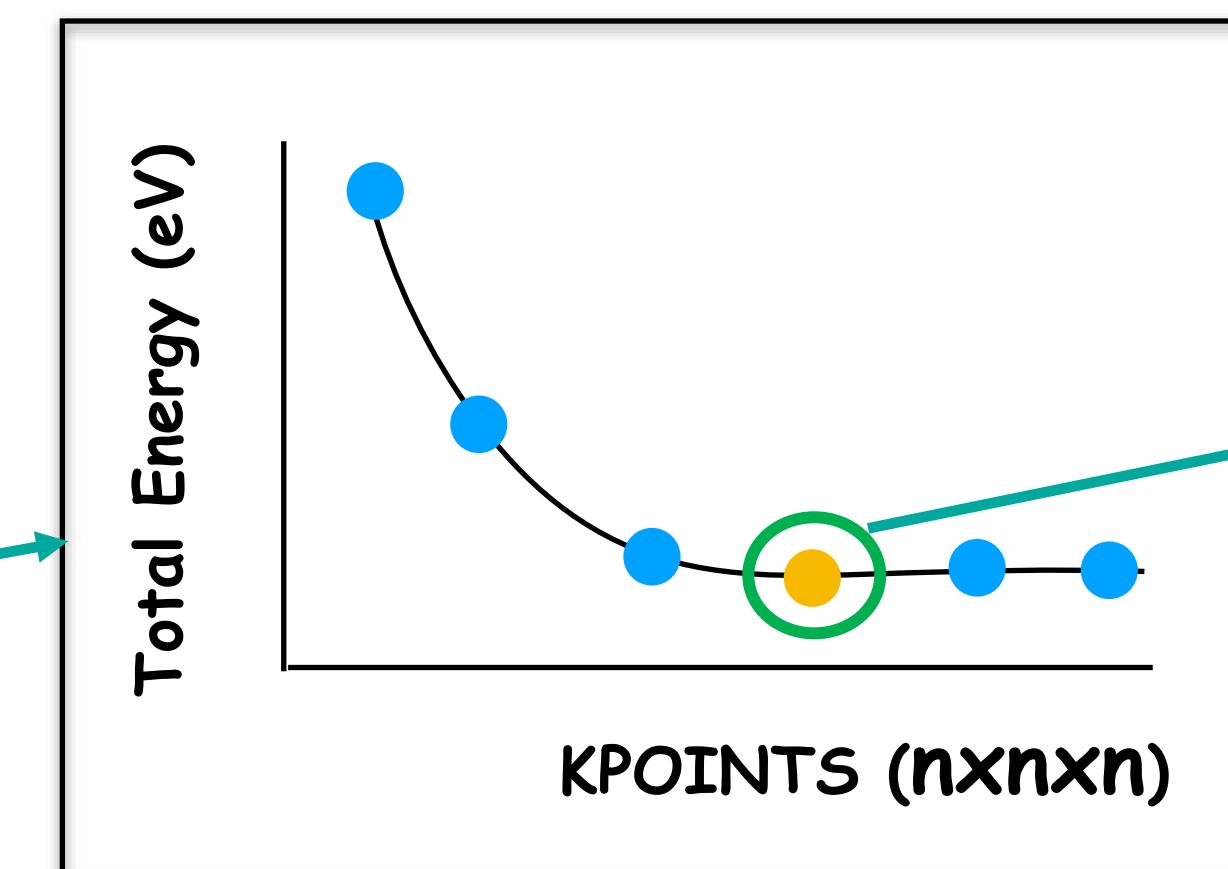
Modify job script to run in a loop

```
#!/bin/bash  
#PBS -N MM747_kc  
#PBS -q little  
#PBS -l nodes=1:ppn=4  
#PBS -V  
  
cd $PBS_O_WORKDIR  
  
for k in {7..14..1}; do  
cat KPOINTS_0 | sed "s/<k>/$k/g" > KPOINTS  
mkdir $k  
cp INCAR KPOINTS POSCAR POTCAR $k/.  
cd ./${k}  
mpirun -n 4 -N 4 /home/amrb/vasp/vasp.6.2.0/bin/vasp_std > vasp.out  
cd ..  
done
```

Submit the job:

**qsub job.qsub**

Once the job is completed,  
Plot Total energies with  
respect to KPOINTS values

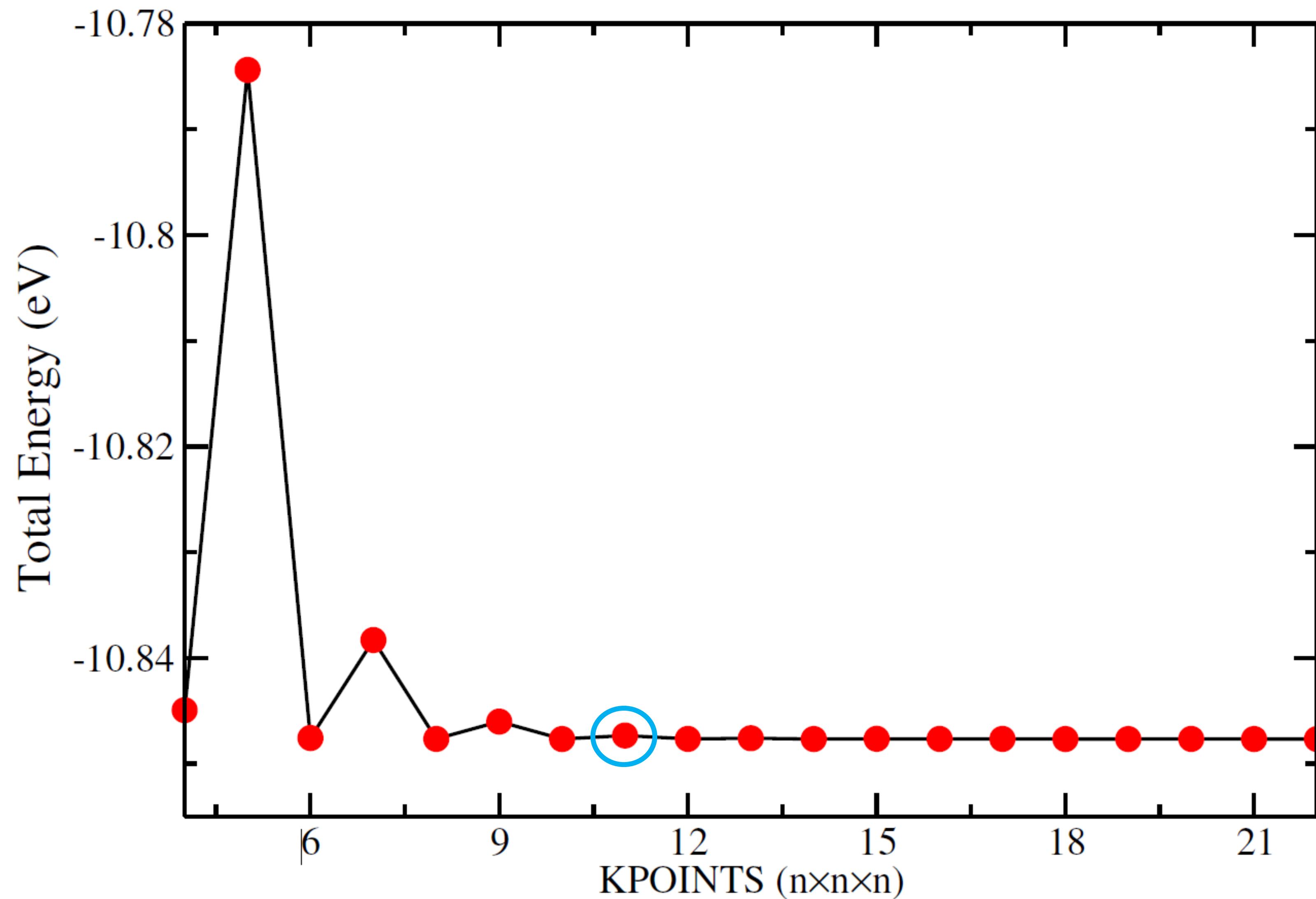


Chosen KPOINTS value  
for further calculations

Computational cost  $\propto$  KPOINTS

# KPOINTS convergence for Si diamond cubic

ESTESODP



# Lattice Convergence

ESTESODP

Create POSCAR\_0 with a place holder

```
System Si
<a>
 2.715 2.715 0.000
 0.000 2.715 2.715
 2.715 0.000 2.715
Si
2
cart
 0.00 0.00 0.00
 0.25 0.25 0.25
```

INCAR, KPOINTS same as previous with converged ENCUT and KPOINTS respectively.

Submit the job: **qsub job.qsub**

Once the job is completed, Plot Total energies with respect to lattice parameter values

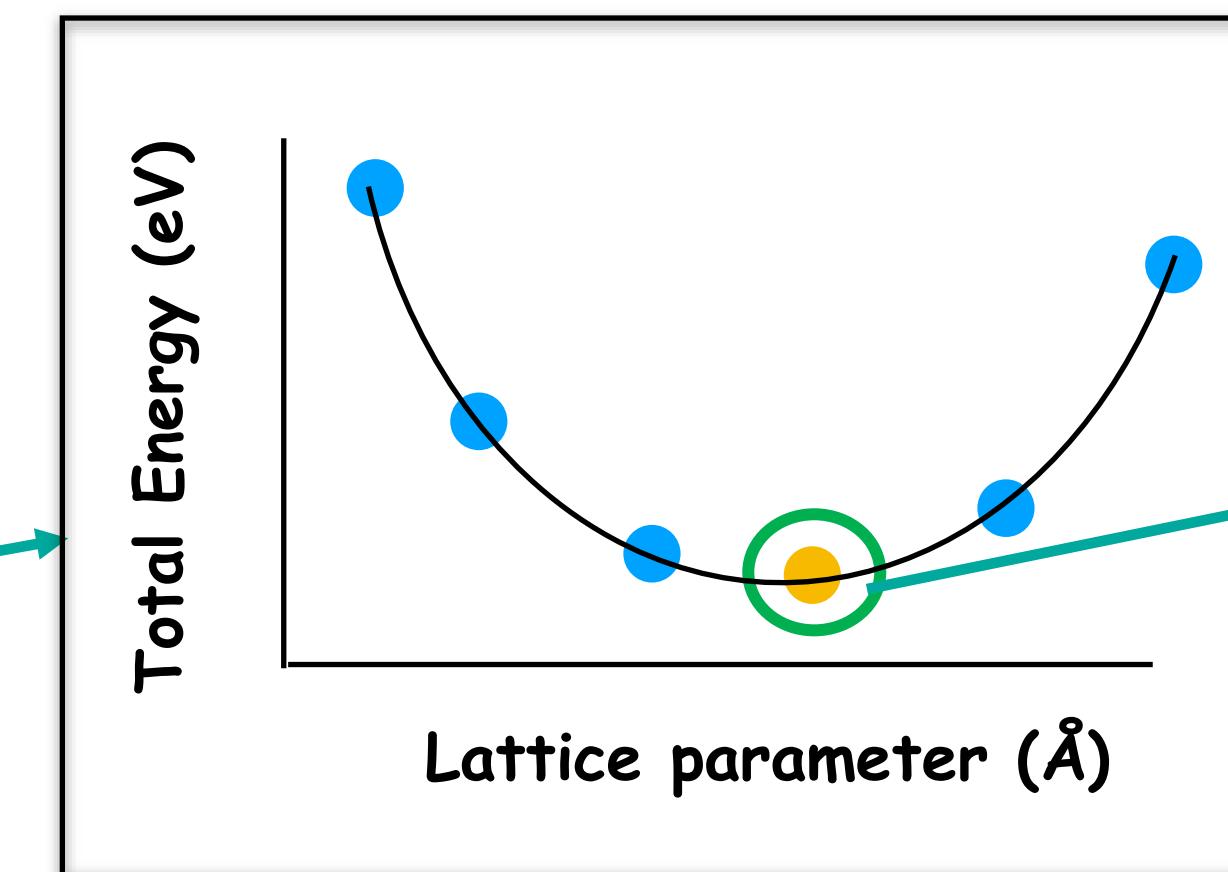
Modify job script to run in a loop

```
#!/bin/bash
#PBS -N MM747_lc
#PBS -q little
#PBS -l nodes=1:ppn=4
#PBS -V

cd $PBS_O_WORKDIR

for a in $(seq 0.90 0.01 1.10); do
cat POSCAR_0 | sed "s/<a>/$a/g" > POSCAR
mkdir $a
cp INCAR KPOINTS POSCAR POTCAR $a/.
cd ./$a
mpirun -n 4 -N 4 /home/amrb/vasp/vasp.6.2.0/bin/vasp_std > vasp.out
cd ..
done
```

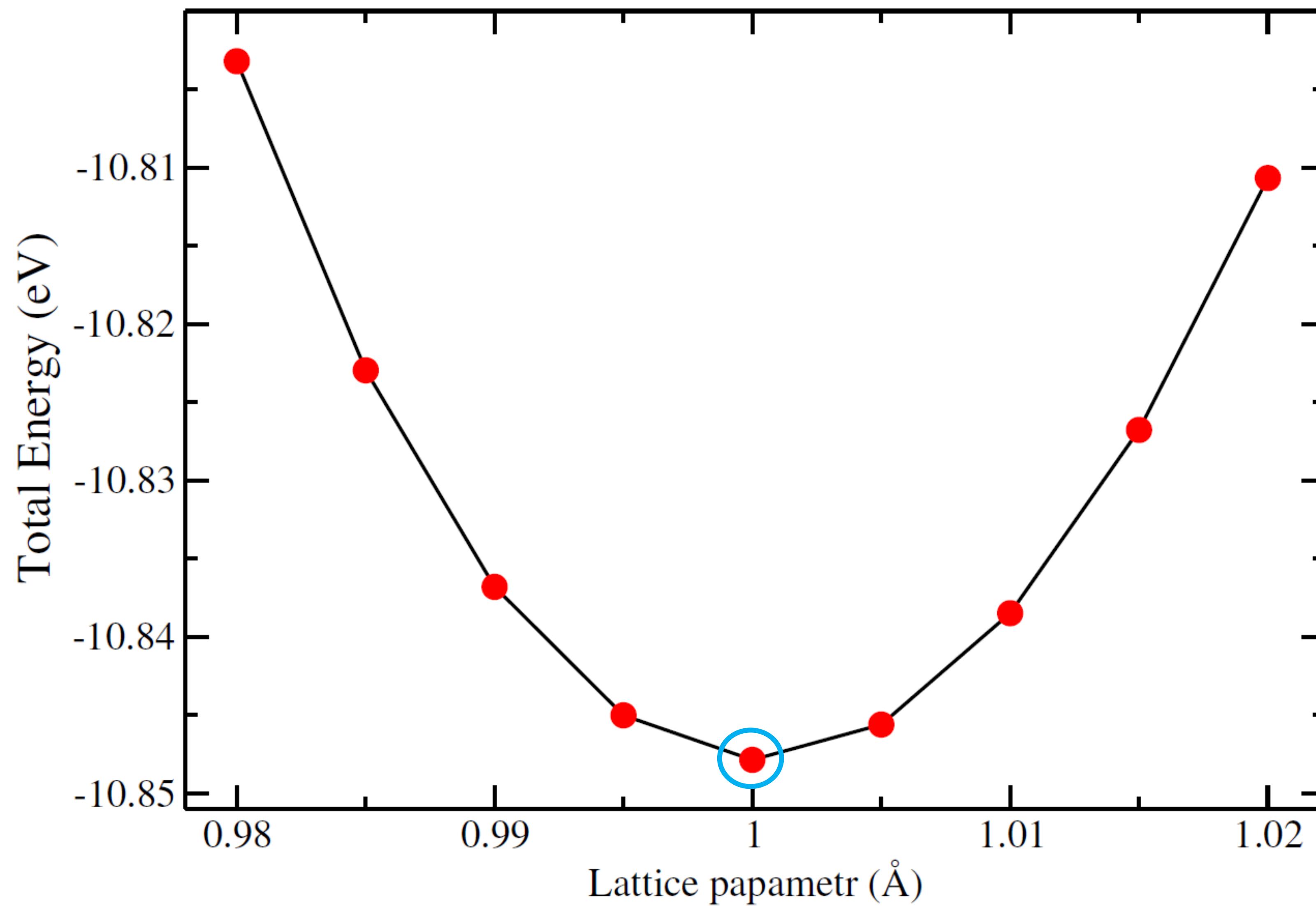
Changing the lattice parameter from -10% to +10%



Chosen Lattice parameter further calculations

# Lattice convergence for Si diamond cubic

ESTESODP



# Structural comparison

ESTESODP

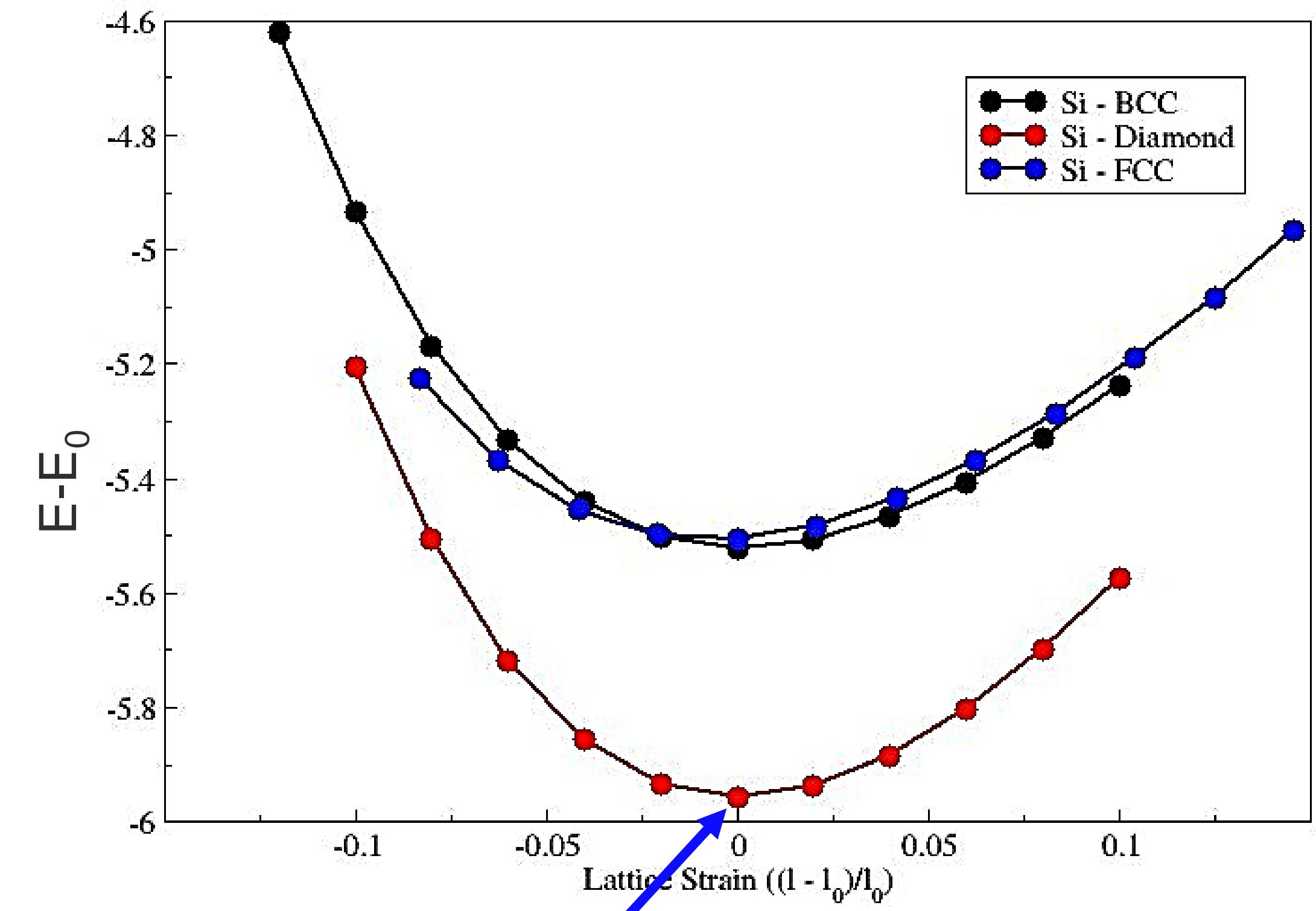
Similar approach is followed for different structures

Convergence tests

Full relaxation with the converged parameters of  
ENCUT and KPOINTS

Total energy comparison

Ground state structure  
determination



Select this structure and do a final relaxation to  
get ground state structure (ISIF = 3)

# Final run to get ground state structure

ESTESODP

Relaxes cell shape  
size and volume

```
ISTART = 0
ISPIN = 1
ENCUT = 400
ISMEAR = 0
SIGMA = 0.05
EDIFF = 1E-06
NSW = 300
IBRION = 2
ISIF = 3
EDIFFG = -2E-03
ISYM = 2
```

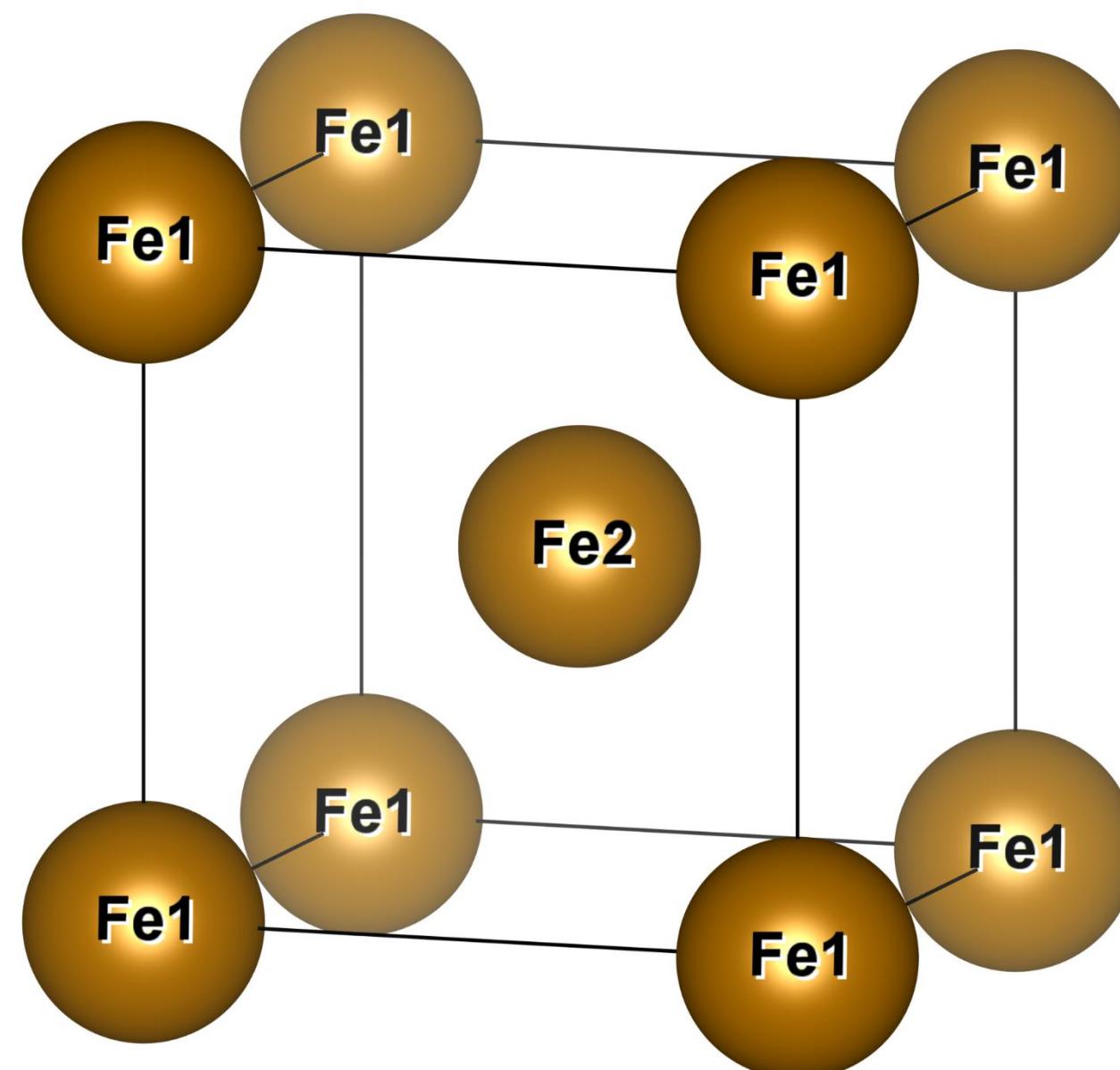
```
Automatic
0
Monkhorst
15 15 15
0. 0. 0.
```

Take the converged  
lattice parameter

A little higher  
KPOINTS for  
accuracy

Total DFT energy = -10.849392 eV  
Lattice parameter = 3.866091292 (Å)

# Calculation for Iron



magnetization (x)

# of ion	s	p	d	tot
-----				
1	-0.012	-0.053	2.247	2.182
2	-0.012	-0.053	2.247	2.182
-----				
tot	-0.024	-0.106	4.495	4.365

```
ISTART = 0
ISPIN = 2
ENCUT = 400
ISMEAR = 0
SIGMA = 0.05
EDIFF = 1E-06
NSW = 300
IBRION = 2
ISIF = 3
EDIFFG = -2E-03
ISYM = 2
```

Converged input values

```
Automatic
0
Monkhorst
15 15 15
0. 0. 0.
```

Take the converged  
lattice parameter

Get input files from this path: [~/input\\_files/Tut-1/Fe](#)

**MAGMOM = NIONS \* X**

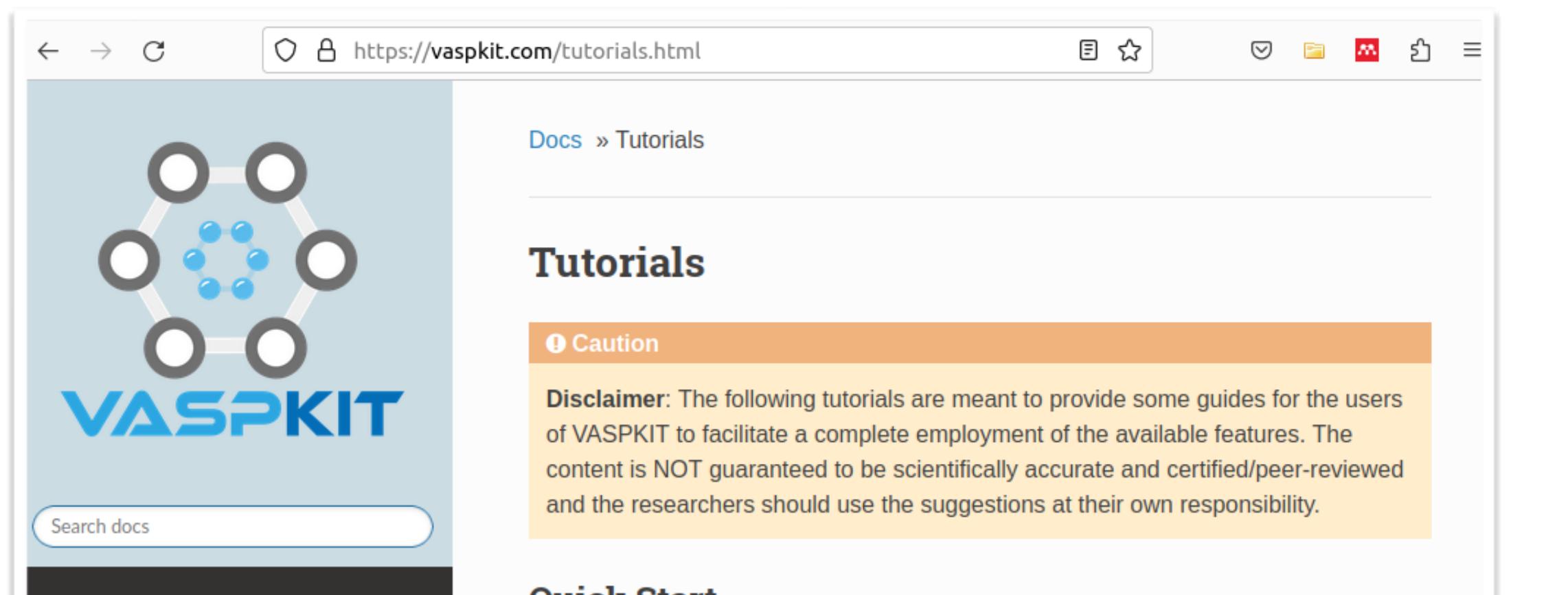
→ Will be taught in Tutorial 3

Total DFT energy = -16.49408993 eV  
Lattice parameter = 2.479461474 (Å)

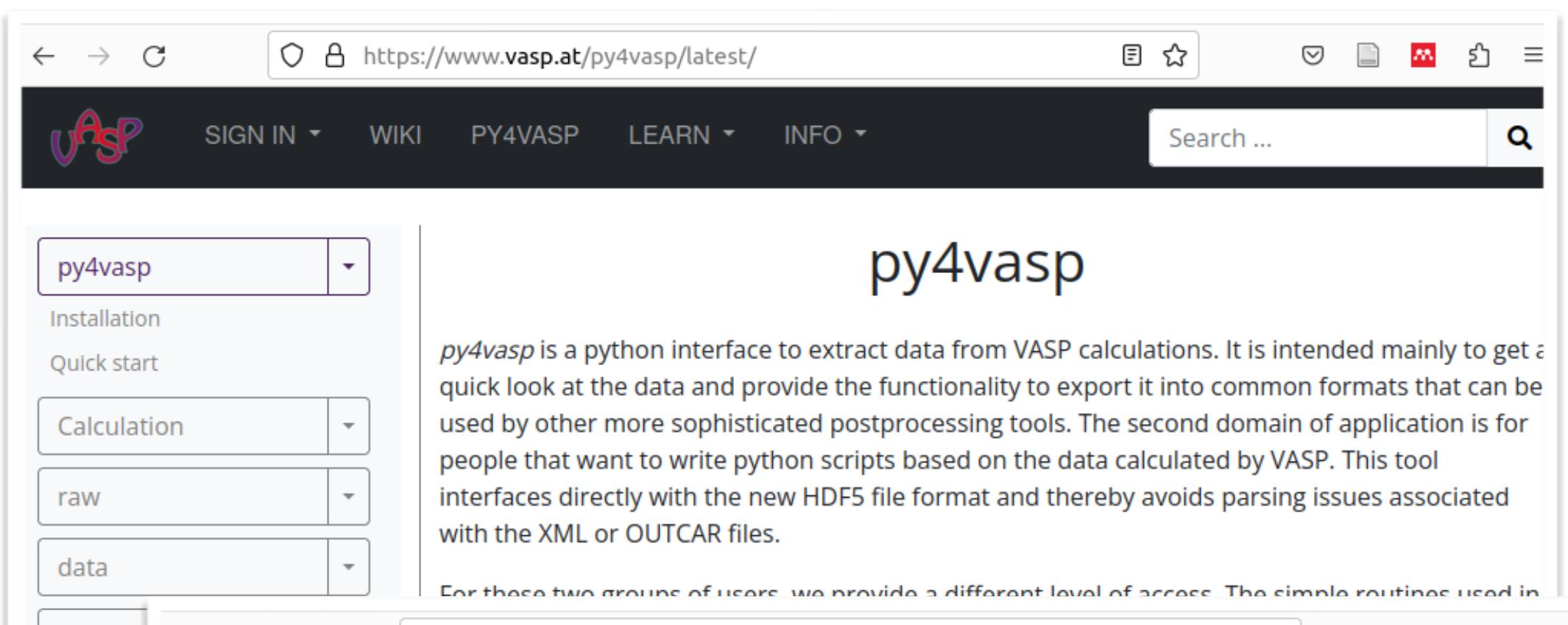
# Artisan's tools

post-processing

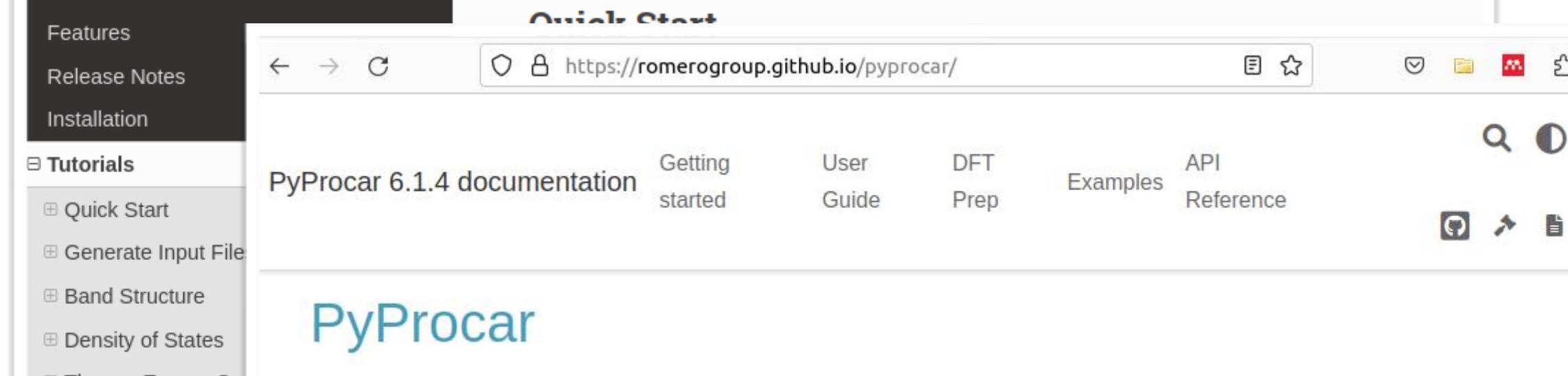
ESTESODP



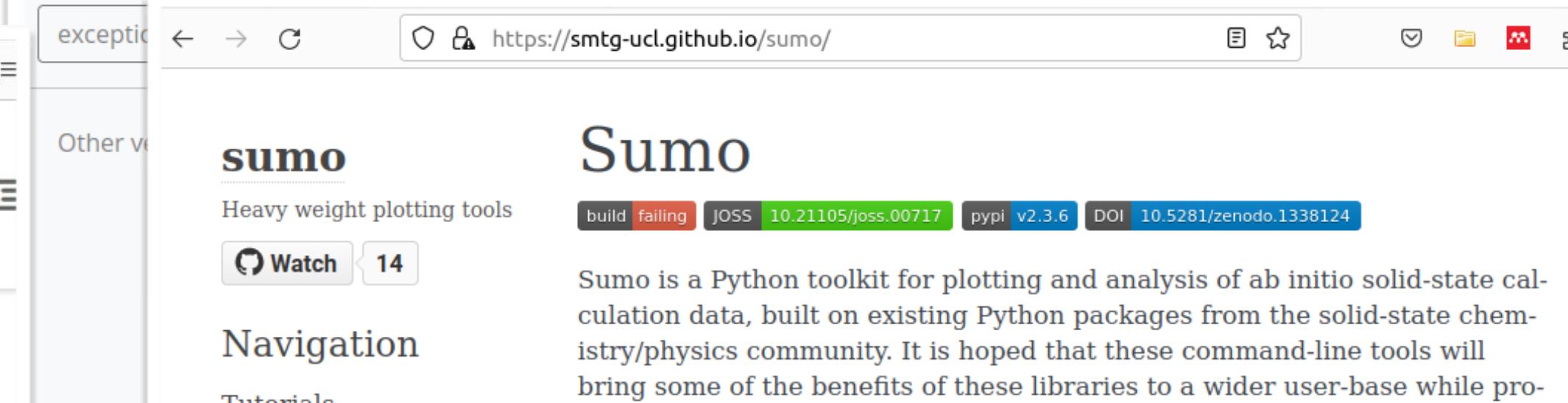
The VASPKIT Tutorials page features a logo of a molecular structure and a search bar. It includes a 'Caution' section with a disclaimer about the accuracy of the tutorials. A 'Quick Start' section provides links to documentation, user guides, and examples.



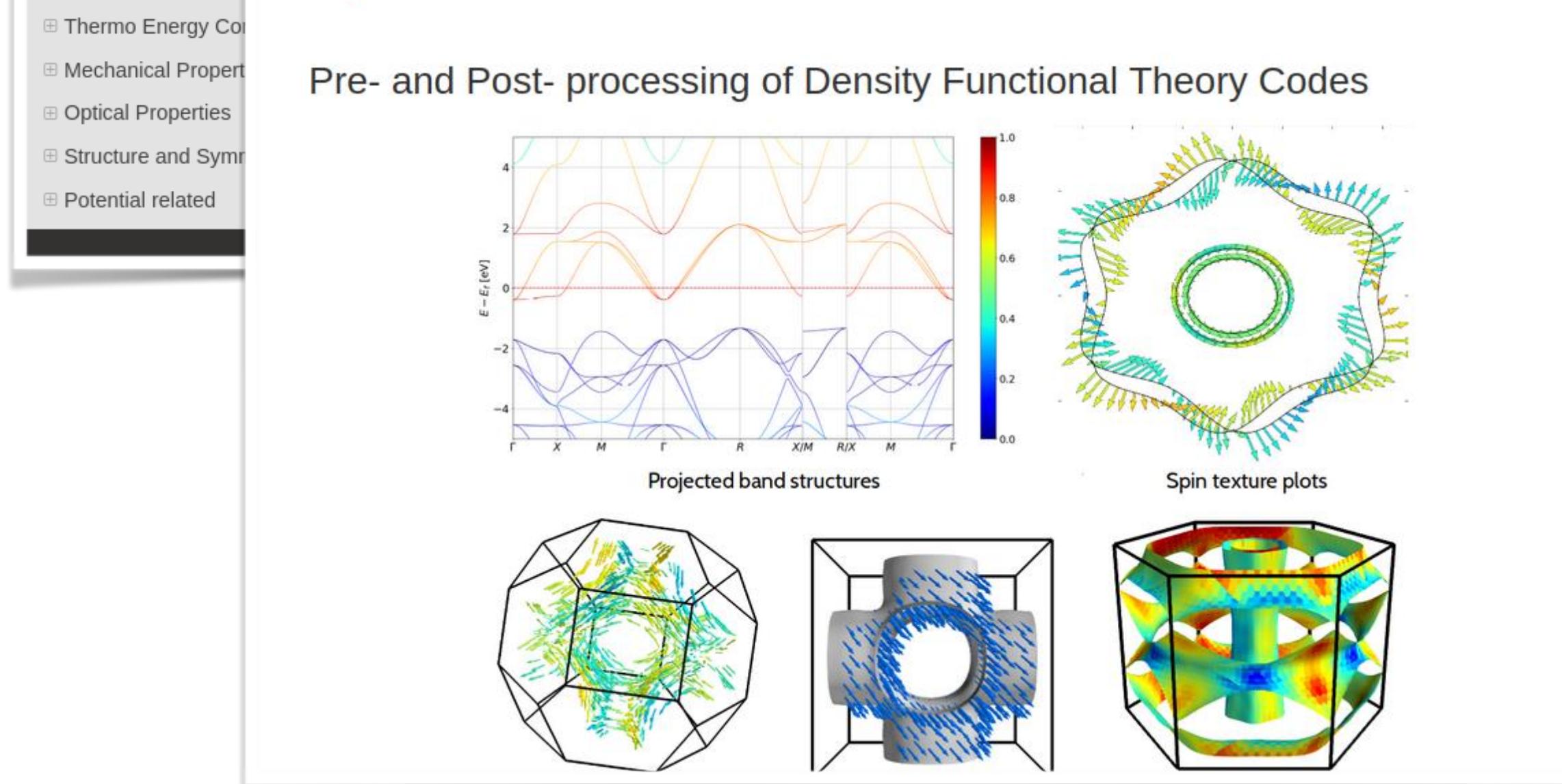
The py4vasp interface shows a dropdown menu for selecting 'py4vasp' or 'vasp'. It includes sections for 'Installation', 'Quick start', 'Calculation', 'raw', and 'data'. A detailed description of py4vasp is provided, noting its use for postprocessing and writing scripts.



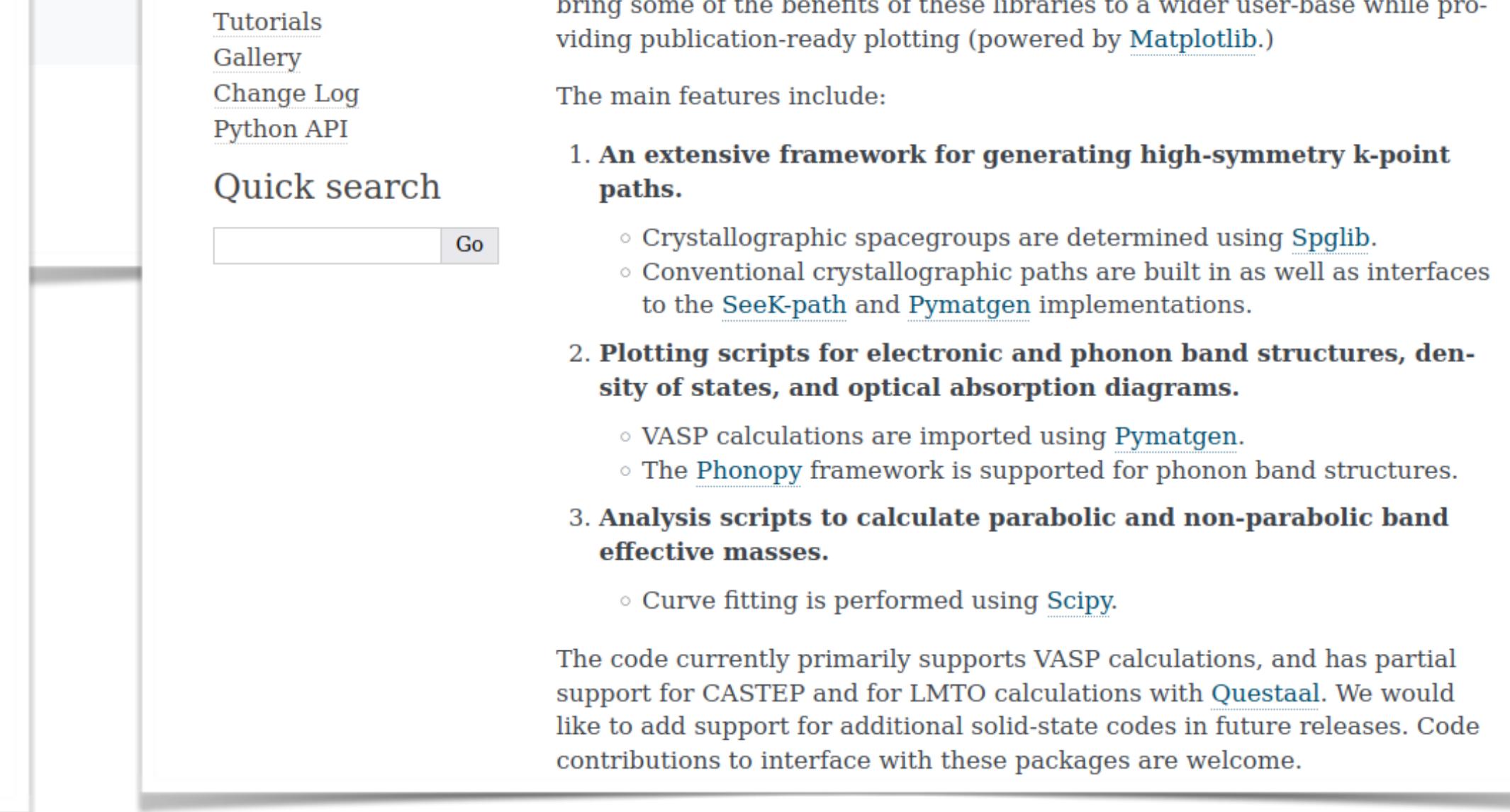
The PyProcar 6.1.4 documentation page includes a navigation bar with links to 'Getting started', 'User Guide', 'DFT Prep', 'Examples', and 'API Reference'. It features a 'Quick Start' section and a 'Tutorials' sidebar with links to various calculation types.



The sumo interface has a 'sumo' header and a 'Heavy weight plotting tools' section. It includes a 'Watch' button with 14 notifications, a 'Navigation' section with links to 'Tutorials', 'Gallery', 'Change Log', and 'Python API', and a 'Quick search' bar.



PyProcar visualization examples include 'Projected band structures' showing energy bands along high-symmetry points, 'Spin texture plots' showing spin density distributions in 3D, and three 3D surface plots illustrating complex crystallographic features.



Sumo features include a 'sumo' header, a 'Heavy weight plotting tools' section, a 'Watch' button with 14 notifications, a 'Navigation' section with links to 'Tutorials', 'Gallery', 'Change Log', and 'Python API', and a 'Quick search' bar. The main features are listed under 'The main features include:'.

- An extensive framework for generating high-symmetry k-point paths.**
  - Crystallographic spacegroups are determined using [Spglib](#).
  - Conventional crystallographic paths are built in as well as interfaces to the [SeeK-path](#) and [Pymatgen](#) implementations.
- Plotting scripts for electronic and phonon band structures, density of states, and optical absorption diagrams.**
  - VASP calculations are imported using [Pymatgen](#).
  - The [Phonopy](#) framework is supported for phonon band structures.
- Analysis scripts to calculate parabolic and non-parabolic band effective masses.**
  - Curve fitting is performed using [Scipy](#).

The code currently primarily supports VASP calculations, and has partial support for CASTEP and for LMTO calculations with [Questaal](#). We would like to add support for additional solid-state codes in future releases. Code contributions to interface with these packages are welcome.

Thank You..!