VASP - Best Practices Seminar

Weine Olovsson
National Supercomputer Centre (NSC), Linköping University
NAISS training, online 3rd Oct 2025

LINKÖPINGS UNIVERSITET

NAISS

https://www.vasp.at/

NAISS training

- Weine Olovsson (NSC) - presentation
- Diana Iusan (UPPMAX) - helper
- Pavlin Mitev (UPPMAX) - helper

https://www.naiss.se/training/

Schedule

https://www.nsc.liu.se/support/Events/VASP_seminar_2025/

Friday 3rd October

10:00 -11:00 Part 1.

11:15 -12:00 Part 2.

12:00 -13:00 L u n c h

13:00 -14:00 Q&A, discussion

- Seminar before lunch
- Questions, discussion, comments -> after lunch
- This presentation -> download .pdf at event page

VASP - Best Practices Seminar

- Basic theory (PAW)
- General considerations …at specific NAISS supercomputers
- Focus on practical aspects of running VASP
- Influential parameters, NPAR/NCORE, ALGO, NSIM, KPAR, …
- Benchmarks, examples
- Common problems … clickable links are underlined

…for utilities & helpful tools, check:
https://www.nsc.liu.se/support/past-events/VASP_workshop_2024/s4-ws2024.pdf

1. Basic Theory (very briefly)

Schrödinger Equation

Time-independent SE

$$H\Psi = E\Psi,$$

Born-Oppenheimer approx.
= 0

$$H = T + T_n + V_{int} + V_{nn} + V_{ext} = -\frac{\hbar^2}{2m_e}\sum_i \nabla_i^2 - \sum_I \frac{\hbar^2}{2M_I}\nabla_I^2 + \\ + \sum_{i\neq j}\frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J \cdot e^2}{|\mathbf{R}_I - \mathbf{R}_J|} - \sum_{i,J}\frac{Z_J \cdot e^2}{|\mathbf{r}_i - \mathbf{R}_J|},$$

solid ~10^{23} particles…

Density Functional Theory (DFT)

Use electron probability density n(**r**) instead of Ψ…

(1) The potential V_{ext} of a system is determined uniquely, except for a constant by the ground state density n(**r**)

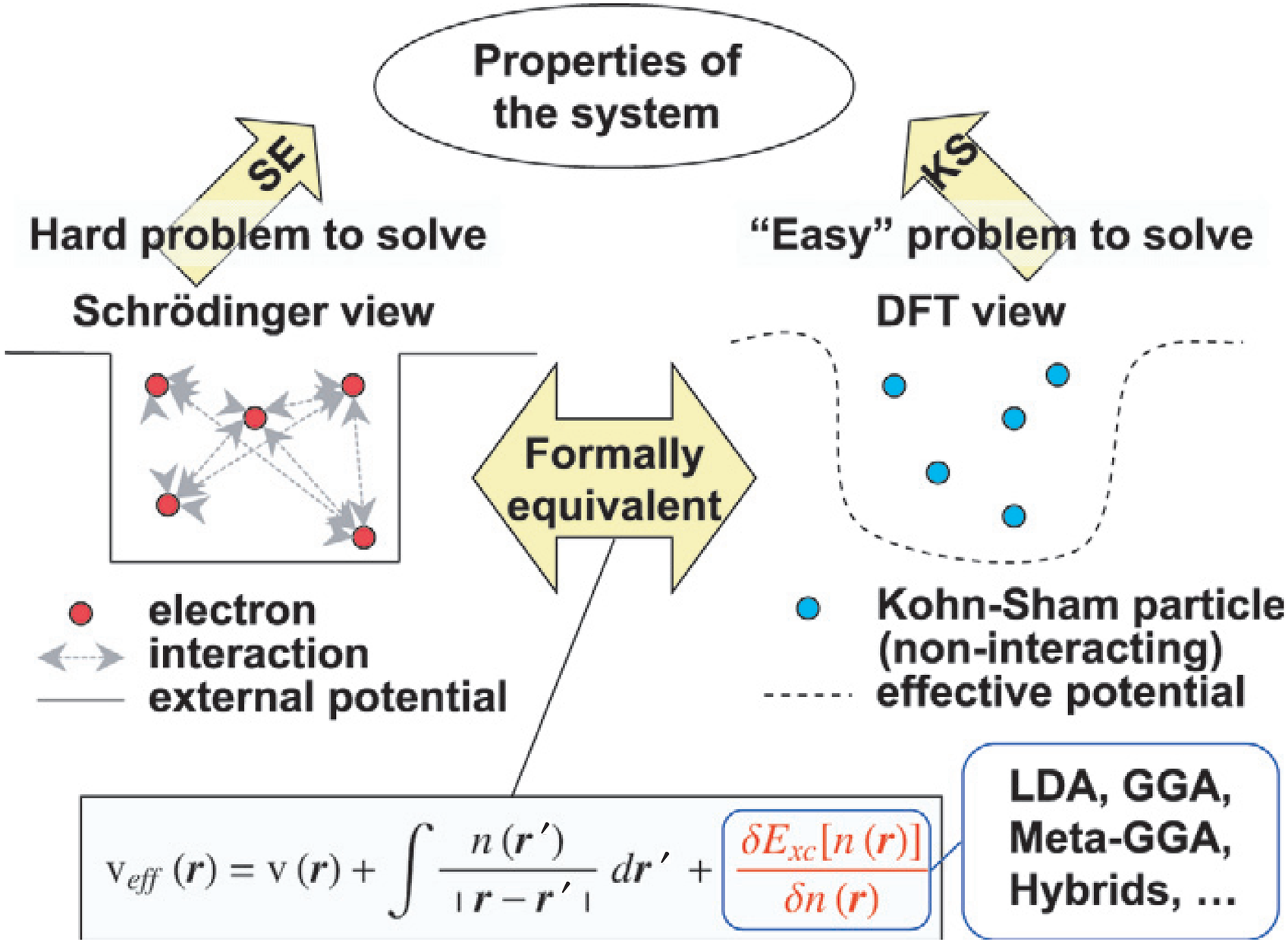
(2) The total energy functional E[n], for a given V_{ext}, assumes its minimal value for the correct electron density n(**r**) of the ground state

Ansatz:

$$E_{KS}[n] = \int d^3r\, V_{ext}(\mathbf{r}) n(\mathbf{r}) + T_s[n] + E_{xc}[n] + \int\int d^3r\, d^3r' \frac{n(\mathbf{r}) n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|},$$

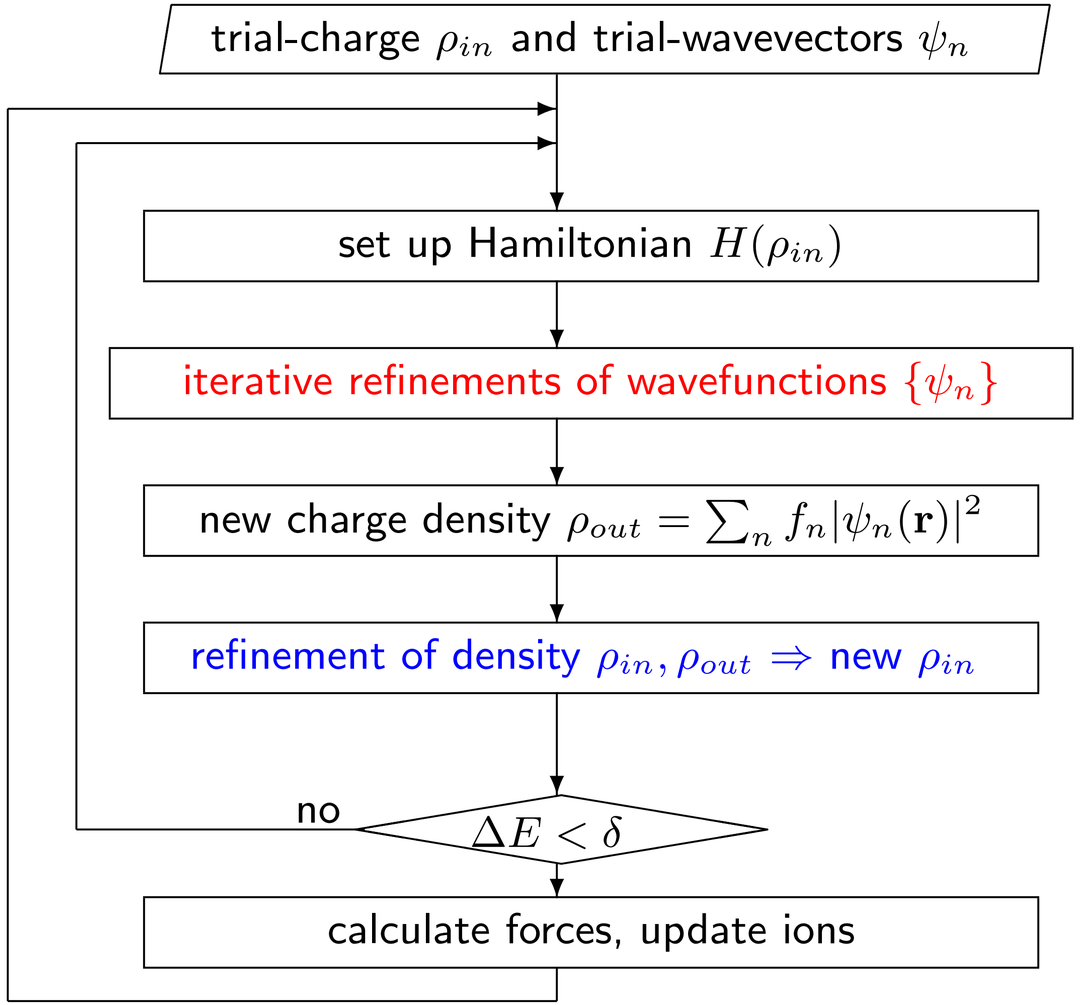
for *independent* electrons (mean field theory)

Hohenberg & Kohn, PRL **136**, B864 (1964)
Kohn & Sham, PRL **140**, A1133 (1965)



Mattsson *et al.*, Modelling Simul. Mater. Sci. Eng. **13**, R1 (2005)

Self-consistent iterations



- two subproblems optimization of $\{\psi_n\}$ and ρ_{in}
- refinement of density: DIIS algorithm

  P. Pulay, Chem. Phys. Lett. 73, 393 (1980)
- refinement of wavefunctions: DIIS or Davidson algorithm

from Marsman: https://www.vasp.at/mmars/day1.pdf

Why PAW?

- **Goal:** both accurate (LAPW) and fast (e.g. USPP) method
- Want to *keep* all-electron (AE) wave function
- Focus on *valence electrons* (frozen core) - chemical bonding
- Fast calculation in *reciprocal space* using FFT (plane waves)
- **Solution:** Projector Augmented Wave (PAW) method

Plane waves & Augmentation

- Rapid wave oscillations close to nucleus

  need too many plane waves!
- Strongly localised states at atoms

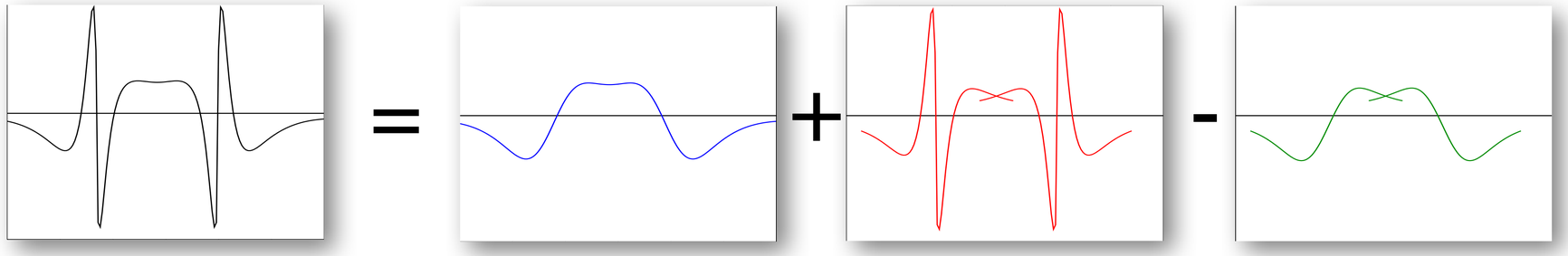
  therefore ->
- Split into *interstitial* and *augmentation* (sphere) regions

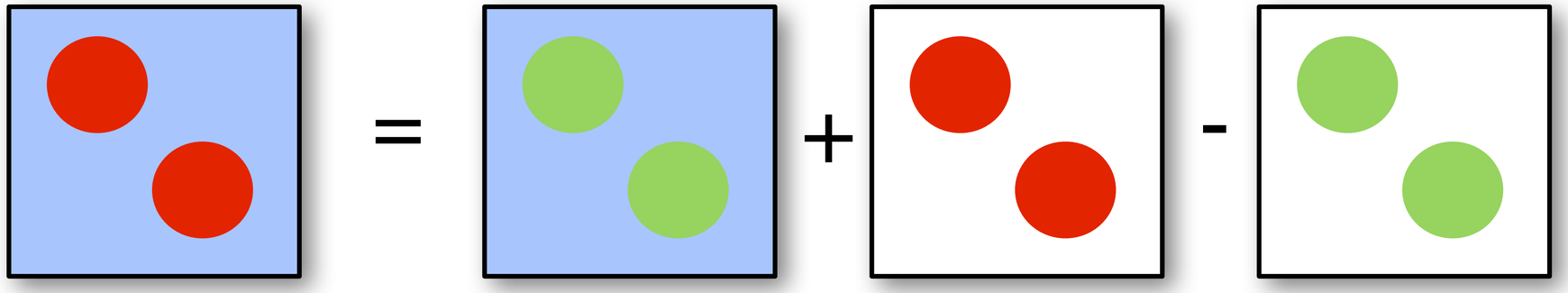
  smooth pw
- No overlap between spheres (one-centre expansion)
- PAW: Energy and potential *independent* wave functions

PAW Augmentation

$$\underbrace{|\psi\rangle}_{\text{all-electron}} = \overbrace{|\tilde{\psi}\rangle}^{\text{pseudo}} + \overbrace{|\psi^1\rangle}^{\text{1-center, all-el.}} - \overbrace{|\tilde{\psi}^1\rangle}^{\text{1-center, pseudo}}$$

$$\sum_\alpha |\phi_\alpha\rangle\langle\tilde{p}_\alpha|\tilde{\psi}\rangle \qquad \sum_\alpha |\tilde{\phi}_\alpha\rangle\langle\tilde{p}_\alpha|\tilde{\psi}\rangle$$





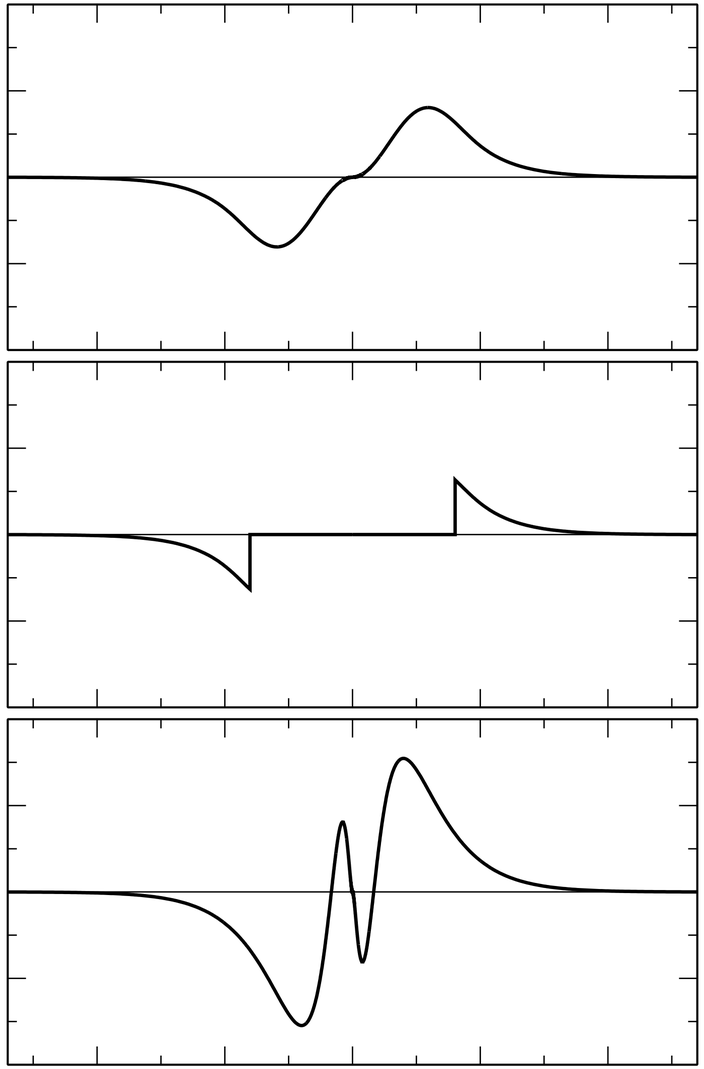
from Blöchl: http://www2.pt.tu-clausthal.de/atp/downloads/lyngby2_paw.pdf

PAW Augmentation

$$|\widetilde{\psi}_n\rangle$$

$$|\widetilde{\psi}_n\rangle - \sum_i |\widetilde{\phi}_i\rangle\langle\widetilde{p}_i|\widetilde{\psi}_n\rangle$$

$$|\widetilde{\psi}_n\rangle - \sum_i |\widetilde{\phi}_i\rangle\langle\widetilde{p}_i|\widetilde{\psi}_n\rangle + \sum_i |\phi_i\rangle\langle\widetilde{p}_i|\widetilde{\psi}_n\rangle$$

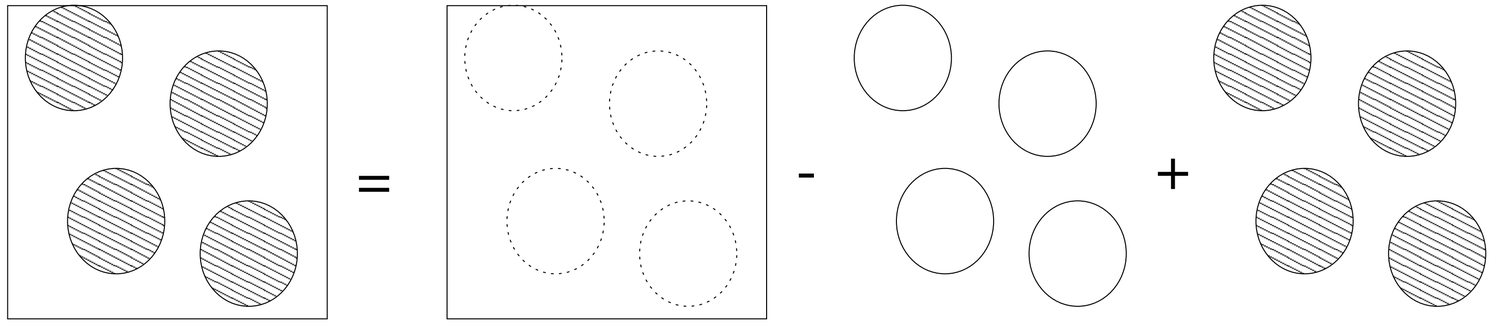


from Marsman: https://www.vasp.at/mmars/day1.pdf

PAW Augmentation

- Character of wavefunction: $c_{lm\epsilon} = \langle \tilde{p}_{lm\epsilon} | \tilde{\psi}_n \rangle$

$$|\psi_n\rangle = |\tilde{\psi}_n\rangle - \sum |\tilde{\phi}_{lm\epsilon}\rangle c_{lm\epsilon} + \sum |\phi_{lm\epsilon}\rangle c_{lm\epsilon}$$



- Same trick works for
  - Wavefunctions
  - Charge density
  - Kinetic energy
  - Exchange correlation energy
  - Hartree energy

from Marsman: https://www.vasp.at/mmars/day1.pdf

Total Energy

$$E = \tilde{E} + E^1 - \tilde{E}^1$$ three terms

PW grid

$$\tilde{E} = \sum_n f_n \langle \tilde{\psi}_n | -\frac{1}{2}\Delta | \tilde{\psi}_n \rangle + E_{xc}[\tilde{\rho} + \hat{\rho} + \tilde{\rho}_c] + E_H[\tilde{\rho} + \hat{\rho}] + \int v_H[\tilde{\rho}_{Zc}] \left(\tilde{\rho}(\mathbf{r}) + \hat{\rho}(\mathbf{r})\right) d^3\mathbf{r} + U(\mathbf{R}, Z_{\text{ion}})$$

local radial grid

$$\tilde{E}^1 = \sum_{\text{sites}} \left\{ \sum_{(i,j)} \rho_{ij} \langle \tilde{\phi}_i | -\frac{1}{2}\Delta | \tilde{\phi}_j \rangle + \overline{E_{xc}[\tilde{\rho}^1 + \hat{\rho} + \tilde{\rho}_c]} + \overline{E_H[\tilde{\rho}^1 + \hat{\rho}]} + \int_{\Omega_r} v_H[\tilde{\rho}_{Zc}] \left(\tilde{\rho}^1(\mathbf{r}) + \hat{\rho}(\mathbf{r})\right) d^3\mathbf{r} \right\}$$

local radial grid

$$E^1 = \sum_{\text{sites}} \left\{ \sum_{(i,j)} \rho_{ij} \langle \phi_i | -\frac{1}{2}\Delta | \phi_j \rangle + \overline{E_{xc}[\rho^1 + \rho_c]} + \overline{E_H[\rho^1]} + \int_{\Omega_r} v_H[\rho_{Zc}] \rho^1(\mathbf{r})\, d^3\mathbf{r} \right\}$$

from Marsman: https://www.vasp.at/mmars/day1.pdf

What are the approximations?

- Frozen core can be relaxed: Marsman & Kresse, JCP 125, 104101 (2006)
- Plane wave expansion, energy cut-off $\frac{1}{2}|\mathbf{G}+\mathbf{k}|^2 < E_{\text{cutoff}}$
- Partial wave expansion (1-2 per angular momentum)

PAW: Things to note

- *All-electron method* (valence states orthogonal to core)
- Frozen core approximation
- Plane waves: FFT in reciprocal space, fast calculations
- **Forces** from total energy expression
- PAW point-of-view: LAPW *special case*, PP an *approximation*

Accuracy

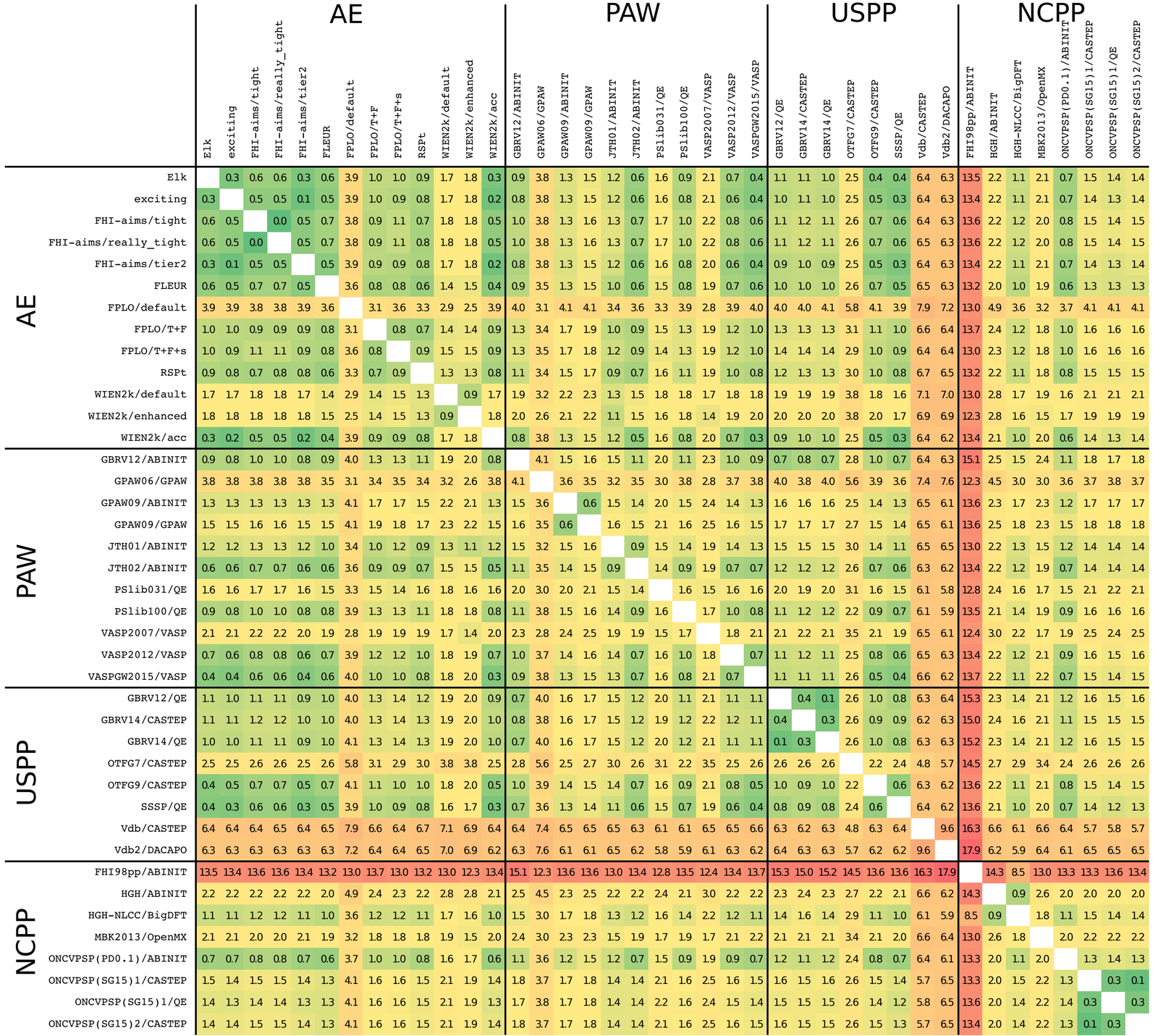
Compare with FPLAPW method (WIEN2k):

$\Delta(\text{PAW})_{(\text{VASP})} = 1.9$ meV/atom

H																	He
0.1																	0.0
Li	Be											B	C	N	O	F	Ne
0.2	0.1											0.3	0.3	10.6	8.3	1.5	0.1
Na	Mg											Al	Si	P	S	Cl	Ar
0.0	0.7											0.3	2.0	3.8	3.3	4.0	0.1
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
0.1	0.2	0.4	0.9	1.3	3.1	1.4	3.4	3.4	2.0	0.4	0.3	0.2	2.4	1.7	1.5	1.5	0.1
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
0.1	0.1	0.5	2.7	7.3	5.5	8.3	2.3	5.4	4.4	4.1	1.4	0.4	0.2	0.1	0.5	0.9	0.1
Cs	Ba	Lu	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
0.3	0.7	4.3	1.2	1.0	3.5	4.3	3.8	1.9	2.5	5.9	0.5	0.4	0.6	0.4	0.4		0.0

Lejaeghere et al. Crit. Rev. Solid State Mat. Sci. 39, 1 (2014)

Lejaeghere et al. Science 351, 1415 (2016)



DFT codes using PAW

- VASP license
- Abinit free
- Quantum Espresso free
- GPAW free
- + more

Importance of good **potential database**

Refs.

- Good presentations by Marsman and Blöchl
- Blöchl PRB **50**, 17953 (1994)
- Blöchl *et al.* https://arxiv.org/abs/cond-mat/0201015v2
- Kresse & Joubert PRB **59**, 1758 (1999)
- Holzwarth *et al.* PRB **55**, 2005 (1997)
- Martin, *Electronic Structure*, Chapter 11.1, 13.2

https://vasp.at/

2. VASP - Basics

LINKÖPINGS UNIVERSITET

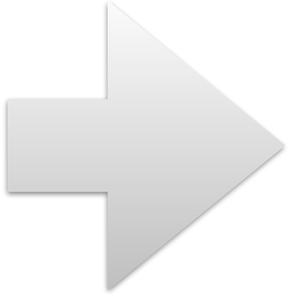
NAISS

VASP - Basics

- Where to find information
  - VASP at different NAISS HPC clusters
- Starting files
- Important parameters
- Input/output
- Examples

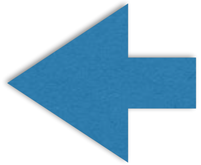
Short background

- Software license (not free!)
- PAW-method
- DFT, **post-DFT** (HSE06, GW, …)
- Born-Oppenheimer Molecular Dynamics
- widely used in Academia/Industry
  - Efforts from Intel & Nvidia for optimization
- 20-25% of Tetralith usage
- VASP6 is available since 2020



https://vasp.at/

Starting advice

- Read the documentation!
- VASP default settings  good starting point
- Caution: “inherited” input files
- Avoid overly messy INCAR…
- Possible differences in installations & versions

refer to respective center webpages / documentation

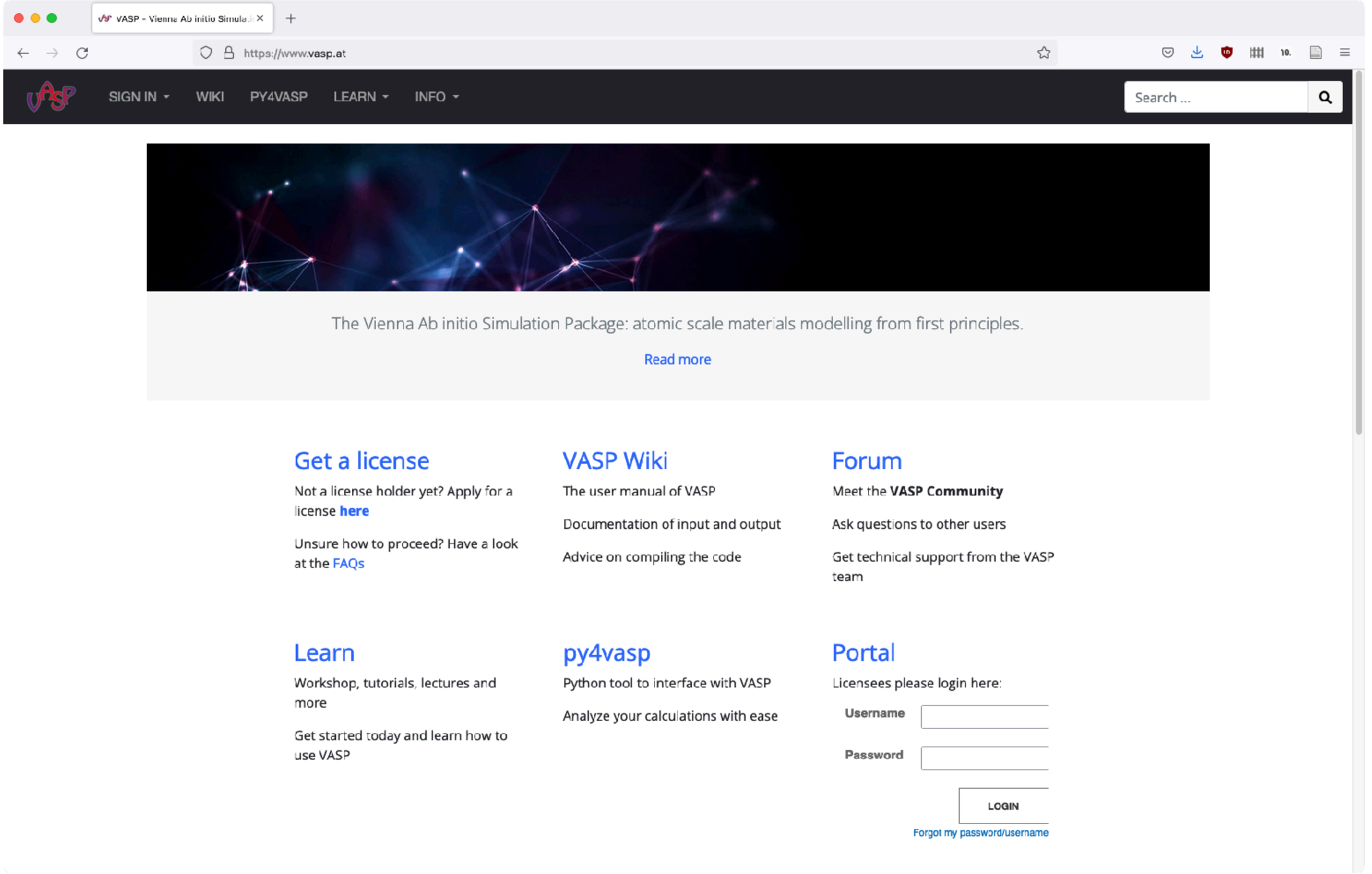
Resources

- Wiki and Manual
  Check in detail!
- Examples, tutorials
- Presentations
- Forum

Find all the links:
https://vasp.at/

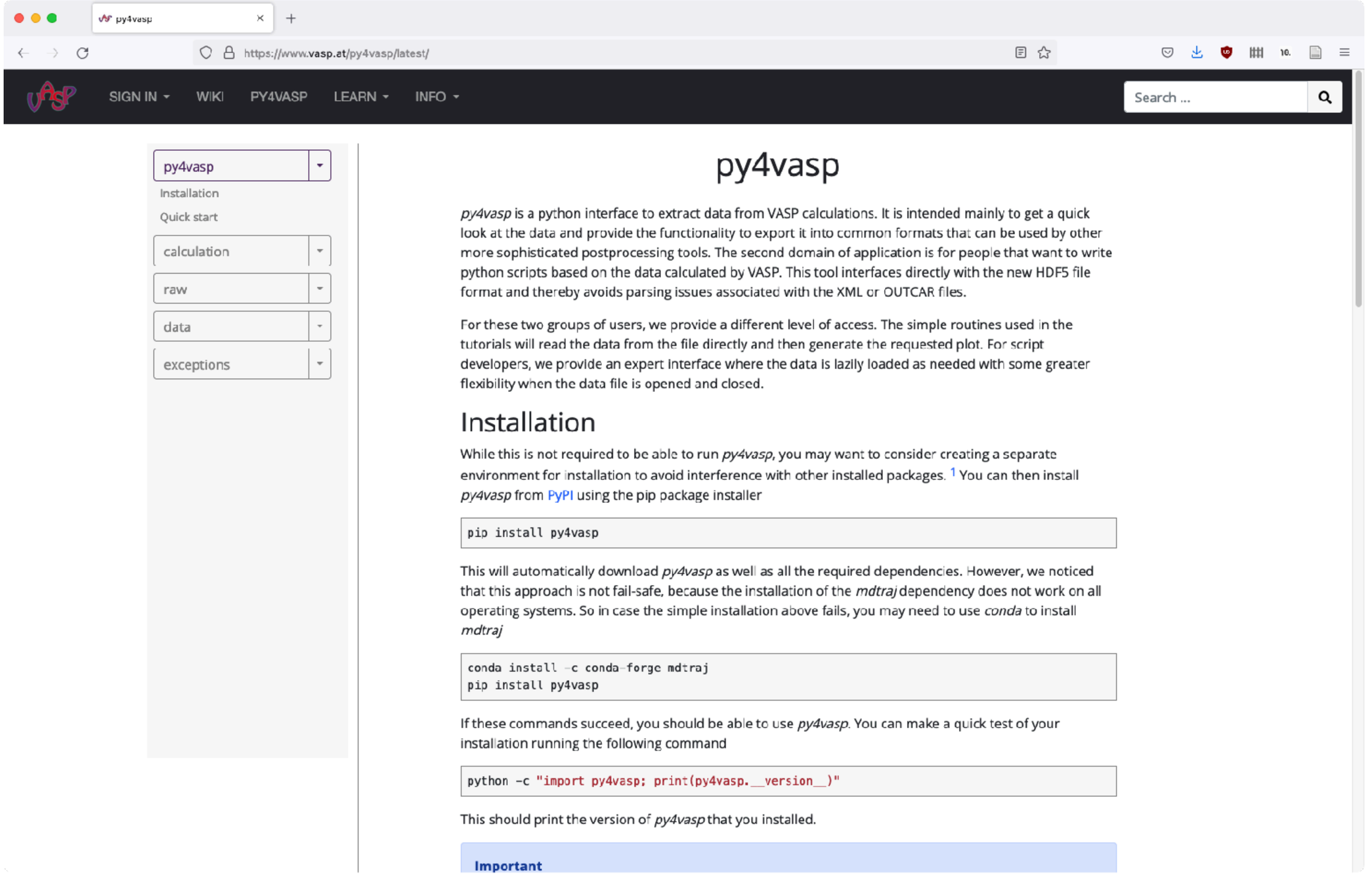
- Also other resources, materials and tools for VASP
- Peter Larsson's old blog at NSC: https://www.nsc.liu.se/~pla/
- Tetralith installations: https://www.nsc.liu.se/software/docs/vasp/
- Dardel installations: https://support.pdc.kth.se/doc/applications/vasp/

Questions / trouble @NAISS clusters? https://supr.naiss.se/support/



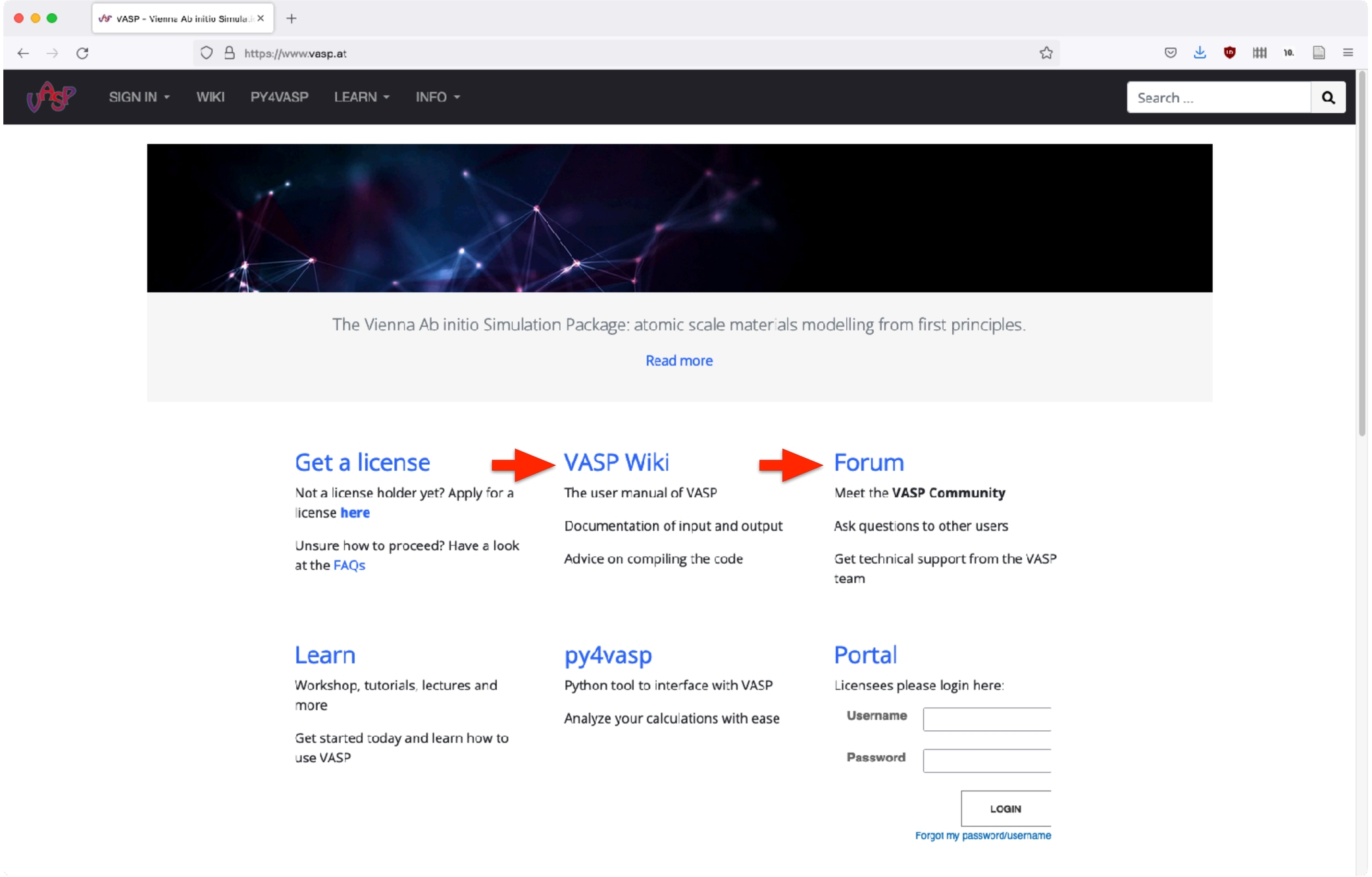
https://www.vasp.at/

…the official VASP webpage



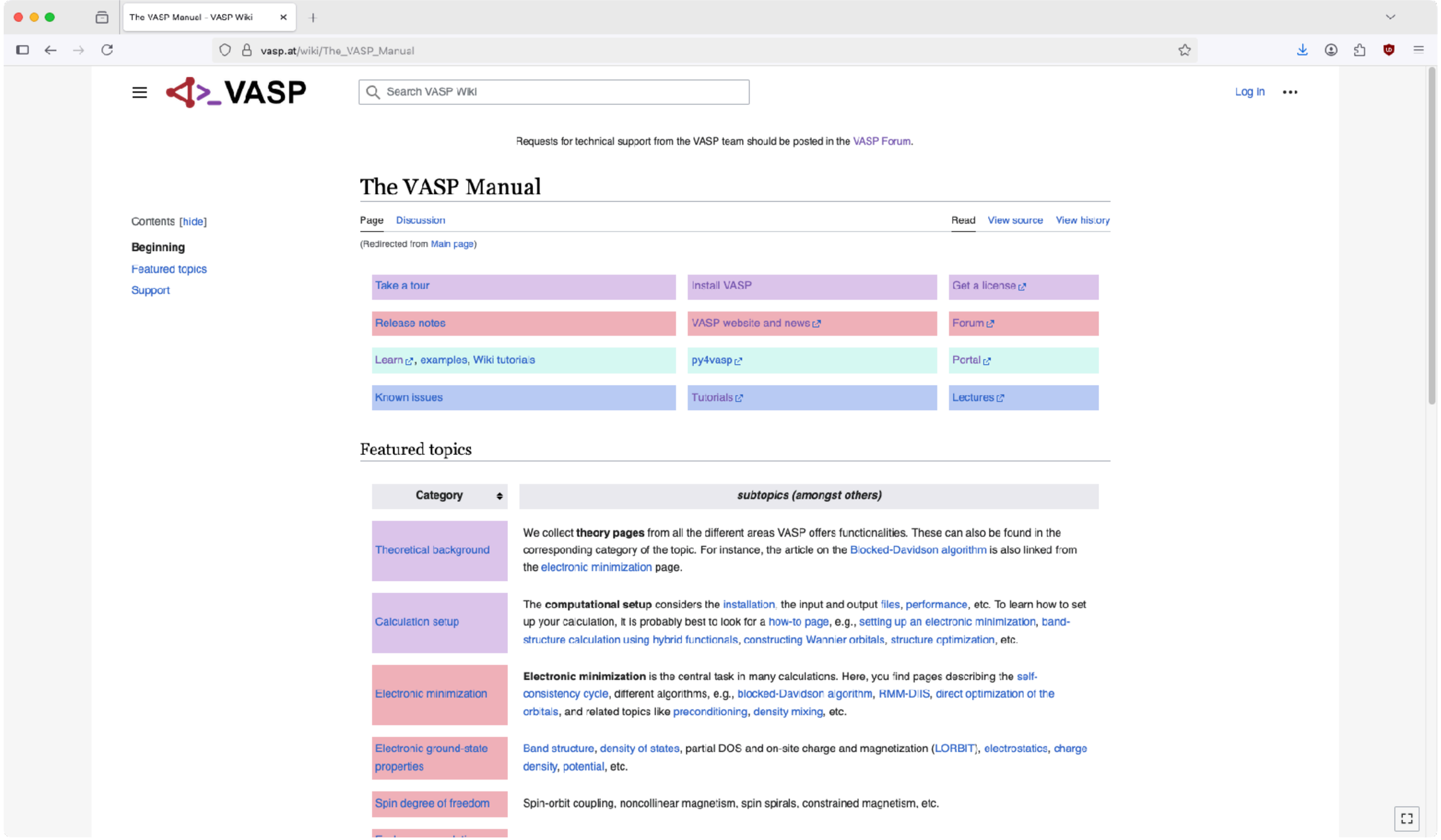
https://www.vasp.at/

…py4vasp useful tool for quick analysis!



https://www.vasp.at/

…wiki important resource & also check forum



https://vasp.at/wiki/The_VASP_Manual

Spin degree of freedom	Spin-orbit coupling, noncollinear magnetism, spin spirals, constrained magnetism, etc.
Exchange-correlation functionals	LDA, GGA, meta-GGA, DFT+U, hybrid functionals, van der Waals functionals.
Symmetry and structure	Crystal symmetry, reciprocal space, surfaces, pair-correlation function for liquids, etc.
Ionic minimization	Structure optimization, ionic-minimization methods, forces, transition states, etc.
Molecular dynamics	Barostats, thermostats, ensembles, etc.
Ensemble properties	Monitoring geometric parameters, pair-correlation function, thermal conductivity, diffusion, etc.
Advanced molecular-dynamics sampling	Interface pinning, constrained molecular dynamics, metadynamics, thermodynamic integration, etc.
Machine-learned force fields	Training and application of force fields.
Phonons	Lattice vibrations, finite differences, phonon dispersion relation.
Electron-phonon interactions	Band-structure renormalization, transport, stochastic sampling.
Response theory	Static and frequency-dependent dielectric properties, Berry phases, spectroscopy (UV, VIS, X-ray, NMR), phonons, etc.
Many-body perturbation theory	ACFDT, BSE, GW, MP2, cRPA.
Localized basis and projection	Obtaining Wannier functions, SCDM, partial DOS and on-site charge and magnetization (LORBIT), Constrained-random-phase approximation
Performance	Parallelization, memory management, profiling, etc.

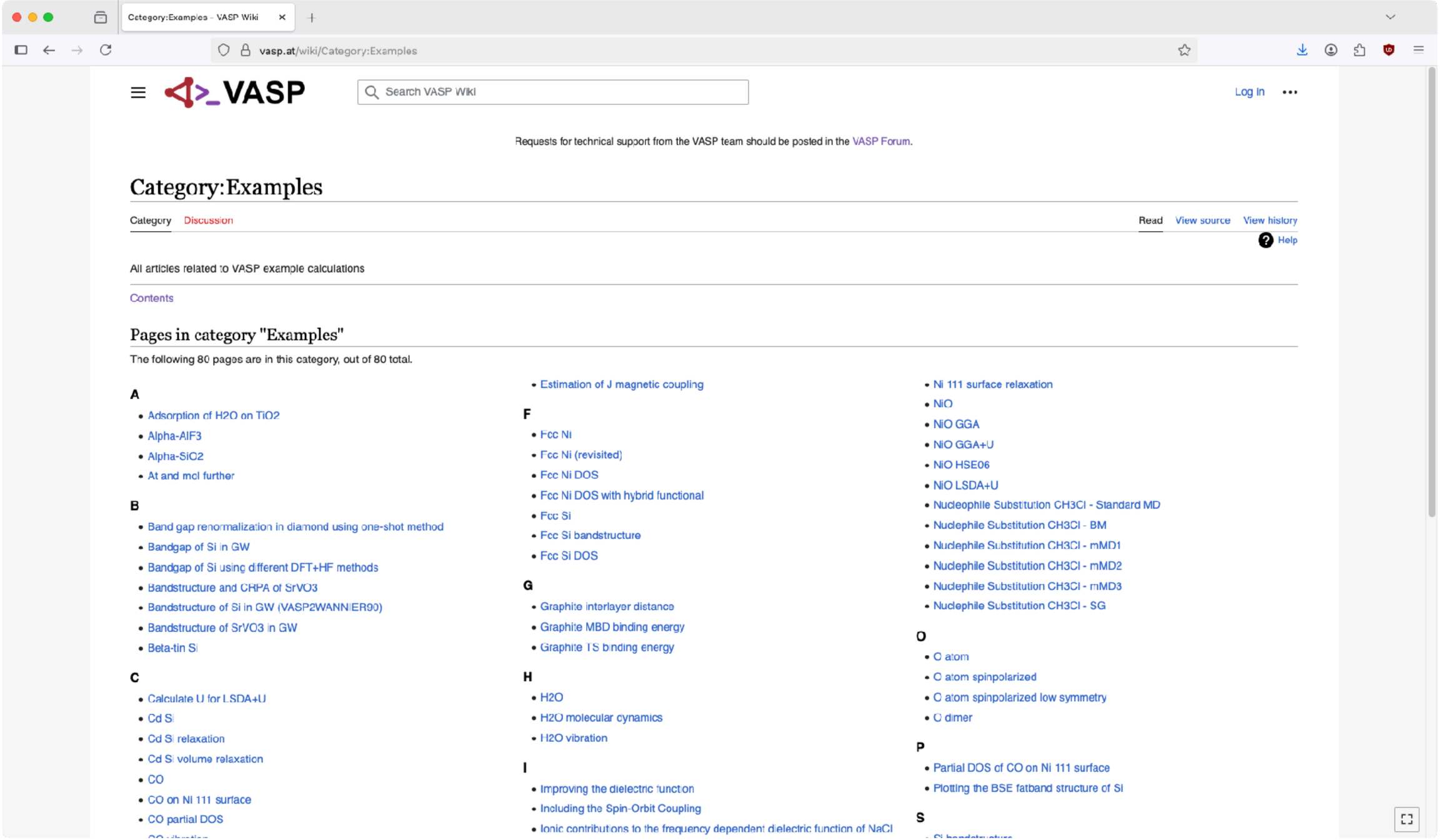
Support

If you have questions or run into trouble, please have a look at the known issues and/or post a question on the VASP Forum.

Mind: We offer support on a courtesy basis only, not as a contractual service.

Back to the top

https://vasp.at/wiki/The_VASP_Manual



https://vasp.at/wiki/Category:Examples

VASP versions & utilities

- **Latest:** 6.5.1 (from March -25)
  - Check center webpages for details!
- wannier90: maximally localized wannier functions
- VTST: transition state tools for VASP
- VASPsol: solvation model for VASP
- Beef: Bayesian error estimation functionals
- constrained relaxation

VASP 6

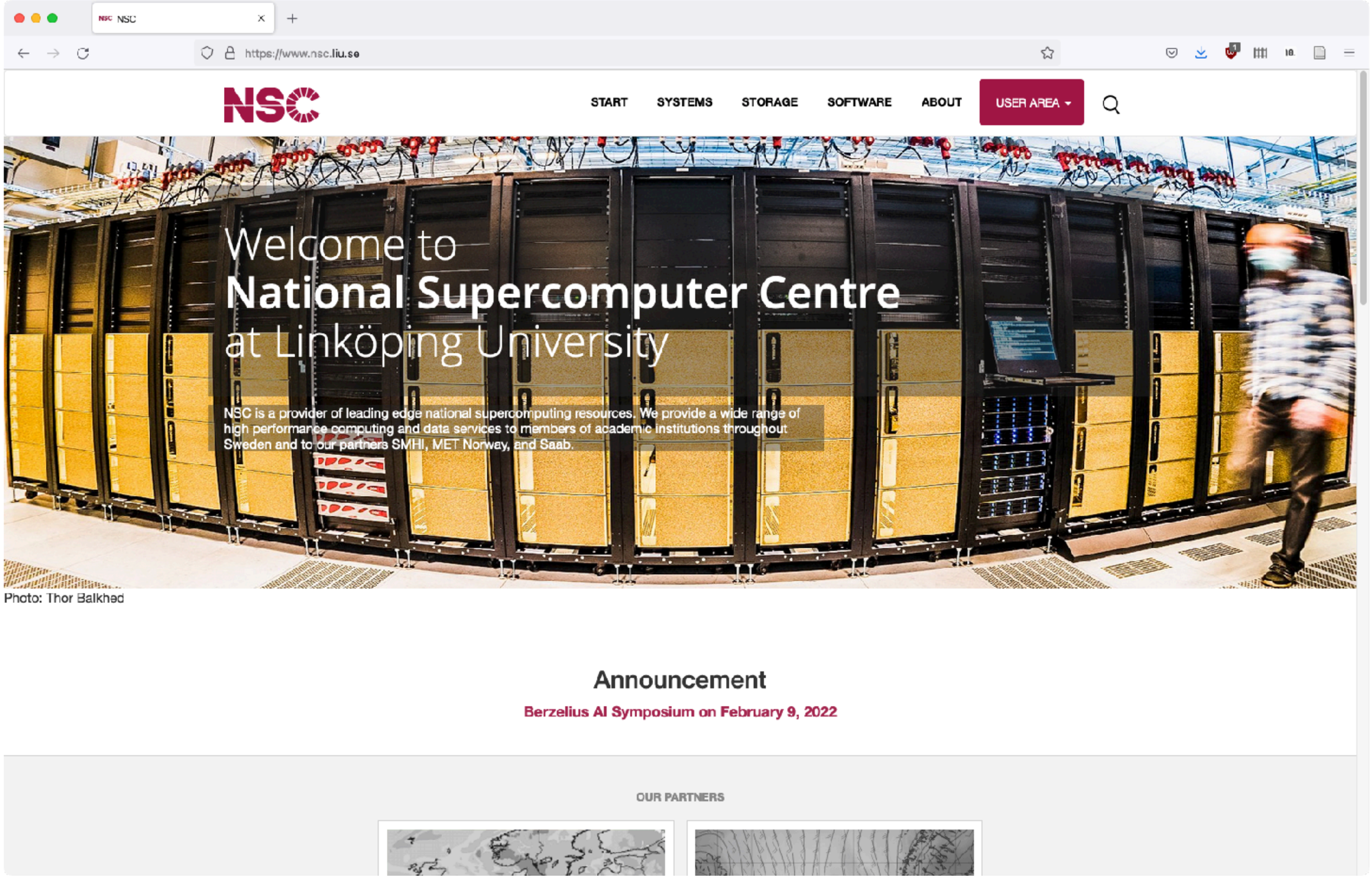
- New license (3y updates), after VASP 5.4.4
- Hybrid OpenMP & MPI parallelization
- OpenACC for GPU
- Cubic scaling RPA and GW
- Electron-phonon coupling using stochastic displacements of atoms
- 6.3: Machine learning force-fields for MD, 6.4 updates -> increased speed (at least 10x noticed)

VASP & HPC centers

- **Tetralith** (**NAISS**) / Sigma, NSC, LiU **installed!**
- **Arrhenius** (**NAISS** & EuroHPC) **<- to be installed!**
- **Dardel** (**NAISS**), PDC, KTH **installed!**
- Other systems/centers (e.g. Kebnekaise, HPC2N) **installed!**
- LUMI (EuroHPC & **NAISS**)
- LEONARDO (EuroHPC) + more EuroHPC

- Check for modules:

```
$ module avail vasp
$ module spider vasp
```



https://www.nsc.liu.se/ Software > Installed software > Tetralith & Sigma software list > VASP

https://www.nsc.liu.se/software/docs/vasp/

VASP6 is available

VASP6 was released in beginning of 2020. This means e.g. that VASP5 license holders will need to update their license in order to access VASP6 installations at NSC. If you have a VASP license 5.4.4, you are probably covered for version up to 6.X.X already, check your license details.

The new features are described in the VASP wiki.

VASP Installations on Tetralith & Sigma

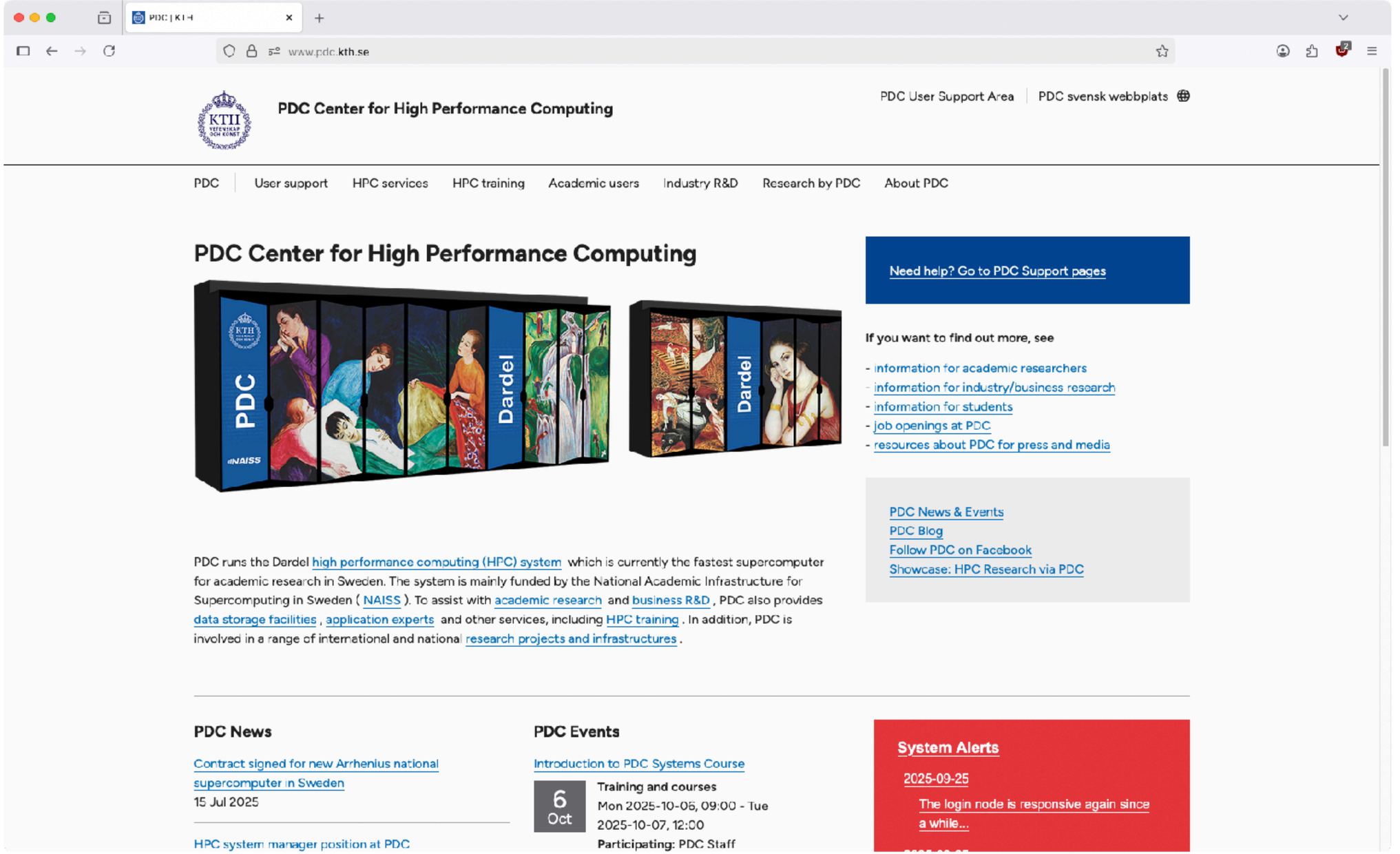
First of all, VASP is licensed software, your name needs to be included on a VASP license in order to use NSC's centrally installed VASP binaries. Read more about how we handle licensing of VASP at NSC.

Some problems which can be encountered running VASP are described at the end of this page.

How to run: quick start

A minimum batch script for running VASP looks like this:

```
#!/bin/bash
#SBATCH -J jobname
#SBATCH -N 4
#SBATCH --ntasks-per-node=32
#SBATCH -t 4:00:00
```



https://www.pdc.kth.se/ User support > Applications > Physics > VASP

https://support.pdc.kth.se/doc/applications/vasp/

Available software
Applications
How to use module to load different softwares into your environment
Bioinformatics
Chemistry
Compilers and languages
Debugging tools
Finite element analysis
Fluid dynamics
Libraries
Mathematics
Molecular dynamics
Performance tools
Physics
Abinit
Cp2k
Elk
Fleur
Libint-cp2k
Libxc
Octopus
Quantum-espresso
Rspt
Spglib
Uppasd
Vasp

Vasp

Installed versions

Resource	Version
Dardel/ cpe24.11	6.5.1-vanilla, 6.5.1-vanilla-xy, 6.5.1-wannier90, 6.4.3-vanilla, 6.4.3-vanilla-xy, 6.4.3-wannier90, 5.4.4-vanilla, 5.4.4-vanilla-xy, 5.4.4-wannier90
Dardel/ cpe23.12	6.3.2-wannier90, 6.4.3-vanilla, 5.4.4-wannier90, 6.2.1-vanilla, 6.3.2-vanilla, 6.4.2-vanilla, 6.2.1-wannier90, 6.2.1-vtst-dftd4, 5.4.4-vanilla, 5.4.4-vtst

General information

The Vienna Ab initio Simulation Package (VASP) is a computer program for atomic scale materials modelling, e.g. electronic structure calculations and quantum-mechanical molecular dynamics, from first principles. For more information see the VASP home page https://vasp.at and the VASP wiki.

Licenses

VASP is not free software and requires a software license. If you want to use VASP please contact us with information of the e-mail address that you have listed in the VASP global portal.

Installed versions
General information

Different types of calculations

- Structural relaxation (different ways)
- Regular E_{tot} scf run using PBE, HSE06, GW, …
- Density of states, bandstructure, charge density, …
- Born-Oppenheimer MD also see tutorial
- Used within a special framework (VTST, …)
- See VASP wiki examples and tutorials

Input files

- INCAR - input parameters
- POSCAR - structure (generate using *e.g.* cif2cell)
- POTCAR - PAW potentials (how to select?)
- KPOINTS - k-mesh (or list)
- + job script

SLURM batch queue system &
settings used by NAISS centers

INCAR parameters

- PREC - “precision”, ENCUT and FFT grids
- ENCUT - plane wave energy cutoff
- ALGO - wf optimisation
- NBANDS - if not set, auto-determined
- NSIM - for RMM-DIIS algorithm (ALGO)
- NCORE or NPAR - bands treated in parallel
- KPAR - k-point parallel

INCAR parameters

accuracy / method

- PREC - “precision”, ENCUT and FFT grids
- ENCUT - plane wave energy cutoff **Completeness of basis-set** **Recommended to set!**
- ALGO - wf optimisation
- NBANDS - if not set, auto-determined **Must be the same for Etot comparison!**

parallel calcs.

- NSIM - for RMM-DIIS algorithm (ALGO)
- NCORE or NPAR - bands treated in parallel
- KPAR - k-point parallel

- We will get back to these settings!

INCAR defaults

- PREC = Normal — Might want “Accurate”
- ENCUT = ? — **Always set!** ENMAX x1.0 - x1.5
- ALGO = Normal — **good tradeoff** Can use “Fast” and “VeryFast”
- NBANDS = ? — can be **overridden** by VASP
  - sometimes extra empty states needed
- NSIM = 4 — Typically OK
- NCORE = 1 — **Adjust** (if not hybrid-functional, HSE06, etc.)
- KPAR = 1 — for k-point parallel calcs.

Will discuss in more detail later on…

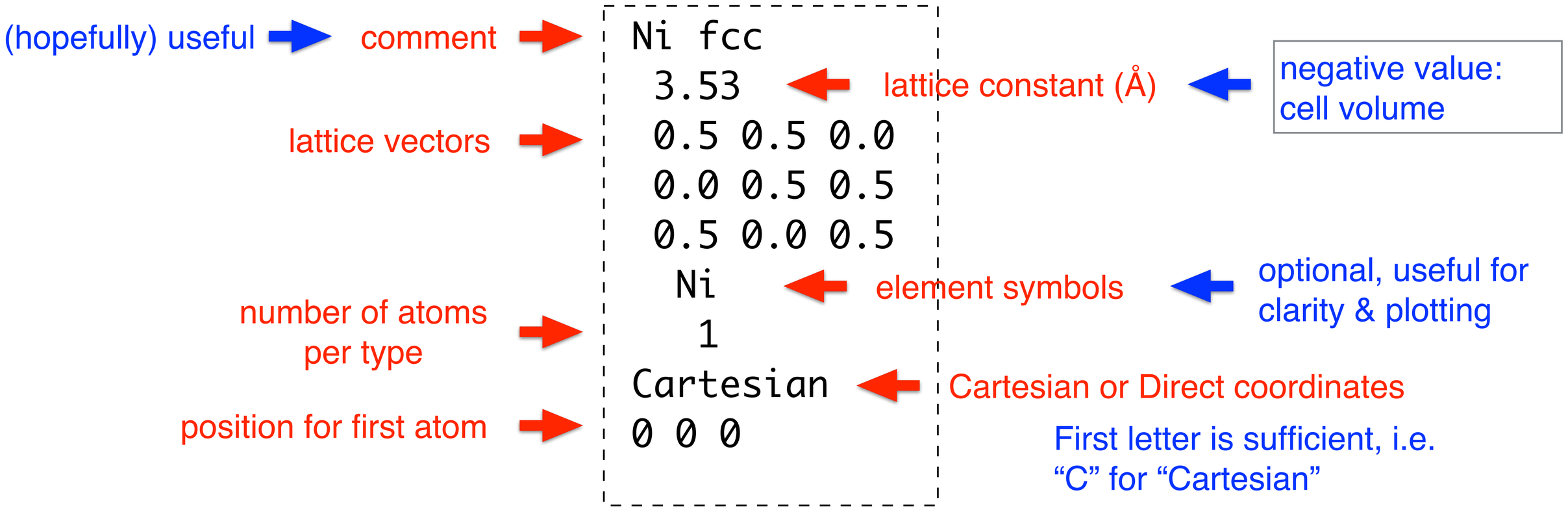
INCAR defaults

in very brief, refer to VASP wiki for details

- NSW = 0 — max ionic steps, also MD steps
- NELM = 60 — max electronic selfconsistency steps
- NELMIN = 2 — min steps. For relaxation/MD set 4-8
- EDIFF = 1E-4 — converge to 4 last digits, sometimes higher accuracy is needed
- EDIFFG = EDIFF x10 — ionic relaxation break condition, if negative value, break if forces < |EDIFFG|
- ISMEAR = 1 — how to treat partial electron occupancy: 1 = metals, 0 = bandgap, -5 = for accurate E_{tot}
- ISPIN = 1 — 2 = spin-polarized calc.
- IBRION = -1 (NSW=-1,0) or 0 — how ions are updated & movded: no update; MD; =2 ionic relaxation

POSCAR

A simple case of fcc Ni, refer to the [VASP wiki example]



Direct coordinates: expressed in terms of the lattice vectors (no lattice constant, scaling)
Cartesian coordinate: expressed as (x,y,z) with the scaling factor included

POSCAR

From one of my own examples, H and Si on Ag(111) surface:

```
100% H on Si on Ag(111)
  10.007900
  1.000000000000000   0.000000000000000   0.000000000000000
  0.500000000000000   0.866025403784439   0.000000000000000
  0.000000000000000   0.000000000000000   4.352531500114909
  H    Si   Ag
  14   14 108
Selective dynamics
Direct
  0.758338000000000   0.052881600000000   0.605945000000000  T   T   T
  0.505244000000000   0.118231000000000   0.605945000000000  T   T   T
  0.050784500000000   0.196017000000000   0.605945000000000  T   T   T
  0.800359000000000   0.251903000000000   0.605945000000000  T   T   T
  0.333333333333333   0.338902000000000   0.605945000000000  T   T   T
  0.114191000000000   0.386244000000000   0.605945000000000  T   T   T
  0.618474000000000   0.498058000000000   0.605945000000000  T   T   T
  0.381262000000000   0.511313000000000   0.605945000000000  T   T   T
  0.883071000000000   0.624684000000000   0.605945000000000  T   T   T
```

Note order of atoms

Relax for different directions

Selective dynamics
always for Direct coord.
T = relax
F = fixed

first H atom

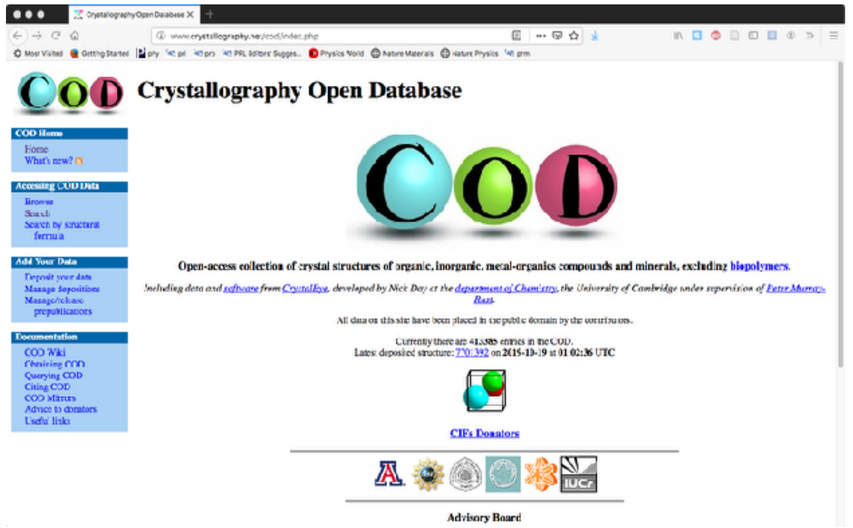
Rest of H, Si & Ag atoms following

POSCAR

Some useful resources:

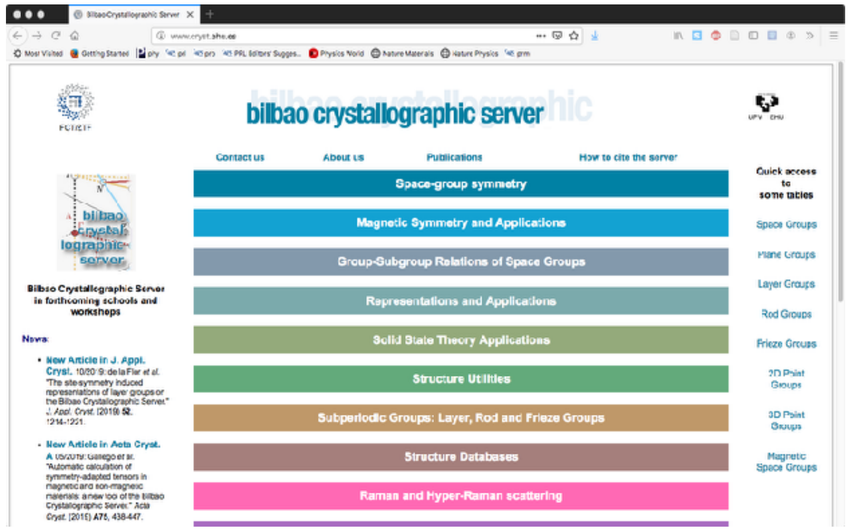
[Crystallography Open Database](#)

Database with published structures from experiment `.cif`



[Bilbao Crystallographic Server](#)

Many Crystallographic tools, e.g. check [BZ of fcc cell](#)



POSCAR

A few examples on how to visualize and/or edit POSCAR:

[Atomic Simulation Environment (ASE)]

Handle structures (and much more) using python scripts, also GUI

[cif2cell]

Versatile script, reads `.cif` saves to many formats including POSCAR - also build supercells

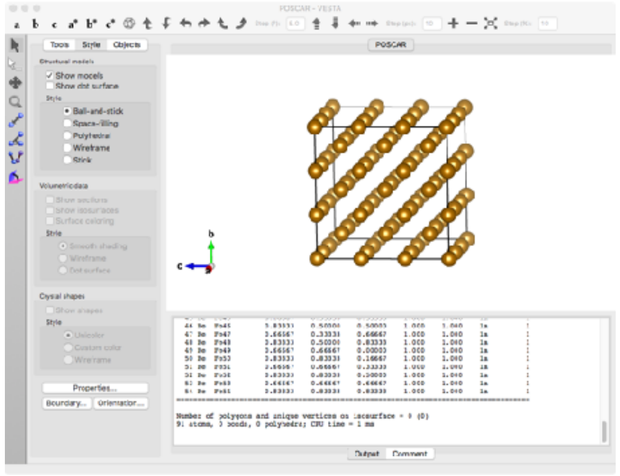
Commercial software:

[NanoLab]

[MedeA]

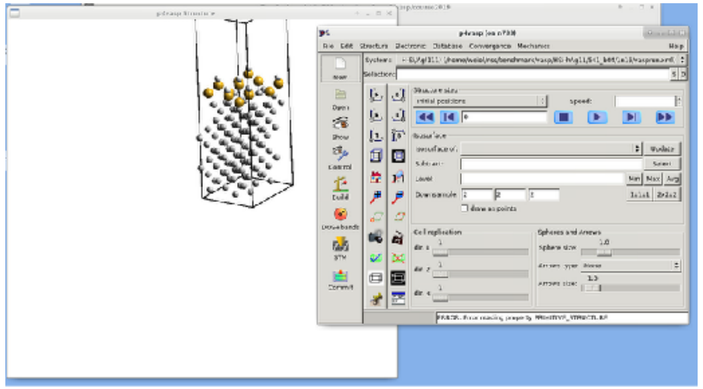
[MaterialsStudio]

[VESTA]



Opens `.cif` displays structure, save as POSCAR

[p4vasp]



Apart from analysis and visualization, also edit POSCAR

POTCAR

PAW potentials - non-trivial to tailor, select with care

```
  PAW_PBE Cu 22Jun2005                                  type, element, date
valence
   11.0000000000000
 parameters from PSCTR are:
    VRHFIN =Cu: d10 p1                                  atomic configuration
XC-type
    LEXCH  = PE
    EATOM  =  1390.9808 eV,  102.2342 Ry

    TITEL  = PAW_PBE Cu 22Jun2005
    LULTRA =        F    use ultrasoft PP ?
    IUNSCR =        1    unscreen: 0-lin 1-nonlin 2-no
    RPACOR =    2.000    partial core radius
    POMASS =   63.546; ZVAL   =   11.000    mass and valenz
    RCORE  =    2.300    outmost cutoff radius
    RWIGS  =    2.200; RWIGS  =    1.164    wigner-seitz radius (au A)
energy cutoff
    ENMAX  =  295.446; ENMIN  =  221.585 eV               smallest energy cutoff
    ICORE  =        3    local potential
    LCOR   =        T    correct aug charges
    LPAW   =        T    paw PP
    EAUG   =  586.980
    DEXC   =    0.000
    RMAX   =    2.344    core radius for proj-oper
    RAUG   =    1.300    factor for augmentation sphere
    RDEP   =    2.302    radius for radial grids
    RDEPT  =    1.771    core radius for aug-charge
```

POTCAR

- Check recommendations, LDA, PBE
- for short bonds: _h
  strong pressure
- for GW: _GW
- States in valence: _sv, _pv, _d
- “soft” (no short bonds): _s
- Where?

Note several choices, e.g.:
Ga, Ga_d, Ga_d_GW,
Ga_GW, Ga_h, Ga_sv_GW

Useful commands:

```
$ grep PAW POTCAR
$ grep ENMAX POTCAR
```

@Tetralith:

```
/software/sse/manual/vasp/POTCARs
/software/sse2/tetralith_el9/manual/vasp/POTCARs
```

KPOINTS

A simple case of fcc Ni, 1 atom

```
k-points
 0
Monkhorst Pack
 11 11 11
 0  0  0
```

comment

0 = automatic generation of mesh

Monkhorst-Pack method (M)

k-point mesh

odd kmesh - includes Γ-point

optional shift of k-mesh

My example, H and Si on Ag(111) surface, 136 atoms

```
Automatic mesh
0
Gamma
2  2  1
0. 0. 0.
```

First letter is sufficient, i.e.
“G” for “Gamma”

Gamma method (G)

- Γ-point included by default
- hexagonal structures only use this!

KPOINTS

For **bandstructure** calculations, provide a list of k-points, see example

```
k-points for bandstructure L-G-X-U K-G
 10
line
Reciprocal
  0.50000  0.50000  0.50000    1
  0.00000  0.00000  0.00000    1

  0.00000  0.00000  0.00000    1
  0.00000  0.50000  0.50000    1

  0.00000  0.50000  0.50000    1
  0.25000  0.62500  0.62500    1

  0.37500  0.7500   0.37500    1
  0.00000  0.00000  0.00000    1
```

comment

k-points per line-segment

k-points per line-segment

Reciprocal / Cartesian

symmetry point + weight

KPOINTS

- Metal - “many” k-pts
- Band gap materials - “few” k-pts
- Unit cell (few atoms) - more k-pts
- Supercell (100s atoms) - few/one, k-pt
- No guarantee for convergence…
- MP method popular, G “safest” to apply
- 1x1x3 cell geometry → 3x3x1 k-mesh

real vs. reciprocal space

real vs. reciprocal space

VASP binaries

- vasp_std - regular version
- vasp_gam - one k-point (Gamma), faster
- vasp_ncl - noncollinear magnetism
- OpenACC GPU binaries, same names
- + modifications

e.g. constrained relaxation

Slurm job script - Tetralith

Walltime limit:
7 days

96 GB RAM / node

Note “mpprun”

```
#!/bin/bash
#SBATCH -A naiss2025-x-yz
#SBATCH -J vasp-run
#SBATCH -t 4:00:00
#SBATCH -N 1 --exclusive

module load VASP/6.5.1.10032025-omp-hpc1-intel-2023a-eb
mpprun vasp
```

Example: running on 1 node (32 cores) @Tetralith

To increase available memory (RAM), reduce cores/node, e.g:

```
#SBATCH --ntasks-per-node=16
#SBATCH --mem=0
```

Alternatively, use “fat” memory nodes (384 GB):

```
#SBATCH -C fat
```

https://www.nsc.liu.se/software/docs/vasp/ *<- more examples!*

Slurm job script - Dardel

Walltime limits:
-p main
1 day max

-p long
7 day max

```
#!/bin/bash
#SBATCH -A naiss2025-x-yz
#SBATCH -J vasp-run
#SBATCH -t 4:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=128
#SBATCH -p main

module load PDC/24.11
module load vasp/6.5.1-vanilla
export OMP_NUM_THREADS=1

srun --hint=nomultithread vasp
```

256 GB RAM / node

Example: running on 1 node (128 cores) @Dardel

Note: queues for larger memory (512GB, 1TB, 2TB) + walltime (7 days) available!

https://support.pdc.kth.se/doc/applications/vasp/ *<- more examples!*

Output files

- OUTCAR - main, detailed output
- OSZICAR - iteration summary
- slurm-***.out - stdout, iteration summary, warnings
- CONTCAR - updated structural data (at finish)

structural relaxation / MD

- XDATCAR - positions at each ionic step
- …

Output files

- DOSCAR - total, partial density of states (DOS)
- CHGCAR - charge density
- WAVECAR - plane wave coefficients (for restart)
- …

output can also be switched off

OSZICAR

Min. algo

Step

Total free Energy

Energy diff.

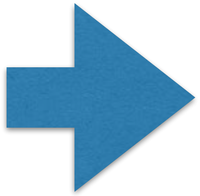
Eigenvalue diff.

Charge density residual vector

```
      N       E                     dE             d eps       ncg     rms          rms(c)
DAV:   1    -0.189343666468E+01   -0.18934E+01   -0.20040E+03   904   0.422E+02
DAV:   2    -0.108926039335E+02   -0.89992E+01   -0.87586E+01  1440   0.554E+01
DAV:   3    -0.109805531666E+02   -0.87949E-01   -0.87949E-01  1208   0.675E+00
DAV:   4    -0.109807517982E+02   -0.19863E-03   -0.19863E-03  1368   0.313E-01
DAV:   5    -0.109807519113E+02   -0.11307E-06   -0.11310E-06  1256   0.684E-03    0.519E+00
DAV:   6    -0.108723496529E+02    0.10840E+00   -0.69164E-02  1064   0.137E+00    0.317E+00
DAV:   7    -0.108218097854E+02    0.50540E-01   -0.13575E-01  1120   0.205E+00    0.163E-01
DAV:   8    -0.108228444695E+02   -0.10347E-02   -0.32972E-03   944   0.419E-01    0.706E-02
DAV:   9    -0.108230614389E+02   -0.21697E-03   -0.22028E-04  1312   0.111E-01    0.557E-02
DAV:  10    -0.108230846187E+02   -0.23180E-04   -0.25743E-05   560   0.381E-02
   1 F= -.10823085E+02 E0= -.10823085E+02  d E =-.431458E-08
```

Final total free energy

Total steps: NELMIN to NELM



Need to check if convergence is reached!

In particular if NELM was reached (default = 60 steps)

Stdout (slurm-***.out)

```
 running on    2 total cores
 distrk:  each k-point on    2 cores,    1 groups
 distr:  one band on    1 cores,    2 groups
 using from now: INCAR
 vasp.5.4.4.18Apr17-6-g9f103f2a35 (build Sep 13 2019 06:30:52) complex

 POSCAR found type information on POSCAR  Si
 POSCAR found :  1 types and       2 ions
 scaLAPACK will be used
 LDA part: xc-table for Pade appr. of Perdew
 POSCAR, INCAR and KPOINTS ok, starting setup
 FFT: planning ...
 WAVECAR not read
 entering main loop
       N       E                     dE             d eps       ncg     rms          rms(c)
DAV:   1    -0.189343666468E+01   -0.18934E+01   -0.20040E+03   904   0.422E+02
DAV:   2    -0.108926039335E+02   -0.89992E+01   -0.87586E+01  1440   0.554E+01
DAV:   3    -0.109805531666E+02   -0.87949E-01   -0.87949E-01  1208   0.675E+00
DAV:   4    -0.109807517982E+02   -0.19863E-03   -0.19863E-03  1368   0.313E-01
DAV:   5    -0.109807519113E+02   -0.11307E-06   -0.11310E-06  1256   0.684E-03    0.519E+00
DAV:   6    -0.108723496529E+02    0.10840E+00   -0.69164E-02  1064   0.137E+00    0.317E+00
DAV:   7    -0.108218097854E+02    0.50540E-01   -0.13575E-01  1120   0.205E+00    0.163E-01
DAV:   8    -0.108228444695E+02   -0.10347E-02   -0.32972E-03   944   0.419E-01    0.706E-02
DAV:   9    -0.108230614389E+02   -0.21697E-03   -0.22028E-04  1312   0.111E-01    0.557E-02
DAV:  10    -0.108230846187E+02   -0.23180E-04   -0.25743E-05   560   0.381E-02
   1 F= -.10823085E+02 E0= -.10823085E+02  d E =-.431458E-08
 writing wavefunctions
```

- Check for warnings!

Warning/advice output

Check stdout (slurm-*.out) for warnings!**

Typical warnings:

Reminder to set (if applicable):
NCORE
typically = used cores/nodes

For high accuracy (default) keep:
LREAL=.FALSE.

```
 -----------------------------------------------------------------------------
|                                                                             |
|           W    W    AA    RRRRR   N    N  II  N    N   GGGG   !!!          |
|           W    W   A  A   R    R  NN   N  II  NN   N  G    G  !!!          |
|           W    W  A    A  R    R  N N  N  II  N N  N  G       !!!          |
|           W WW W  AAAAAA  RRRRR   N  N N  II  N  N N  G  GGG   !           |
|           WW  WW  A    A  R   R   N   NN  II  N   NN  G    G               |
|           W    W  A    A  R    R  N    N  II  N    N   GGGG   !!!          |
|                                                                             |
|      For optimal performance we recommend to set                            |
|        NCORE= 4 - approx SQRT( number of cores)                             |
|      NCORE specifies how many cores store one orbital (NPAR=cpu/NCORE).     |
|      This setting can  greatly improve the performance of VASP for DFT.     |
|      The default,   NCORE=1            might be grossly inefficient         |
|      on modern multi-core architectures or massively parallel machines.     |
|      Do your own testing !!!!                                               |
|      Unfortunately you need to use the default for GW and RPA calculations. |
|      (for HF NCORE is supported but not extensively tested yet)             |
|                                                                             |
 -----------------------------------------------------------------------------

 -----------------------------------------------------------------------------
|                                                                             |
|  ADVICE TO THIS USER RUNNING 'VASP/VAMP'   (HEAR YOUR MASTER'S VOICE ...):  |
|                                                                             |
|      You have a (more or less) 'large supercell' and for larger cells       |
|      it might be more efficient to use real space projection operators      |
|      So try LREAL= Auto  in the INCAR   file.                               |
|      Mind: If you want to do a very accurate calculations keep the          |
|      reciprocal projection scheme          (i.e. LREAL=.FALSE.)             |
|                                                                             |
 -----------------------------------------------------------------------------
```