VASP: Introduction

Weine Olovsson

National Supercomputer Centre (NSC)

@Umeå University, HPC2N, 29th Oct 2019, room UB334

LINKÖPINGS UNIVERSITET

https://www.vasp.at/

Introduction

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

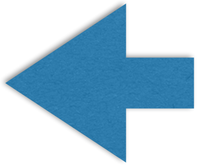
… clickable links are underlined

Short background

- Software license
- 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
- next version is VASP6

http://vasp.at/

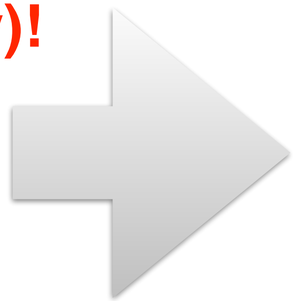
Starting advice

- 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 webpages / documentation

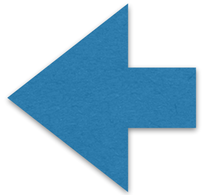
Resources

- Manual
  Read all (really)!
- Wiki
  examples, presentations
- Forum



Find the links:
http://vasp.at/

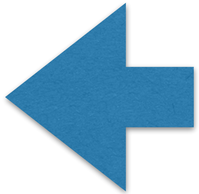
- Peter Larsson's old blog at NSC:
  https://www.nsc.liu.se/~pla/



info & discussion

Questions/trouble? support@nsc.liu.se, support@hpc2n.umu.se, …

VASP at SNIC HPC centers

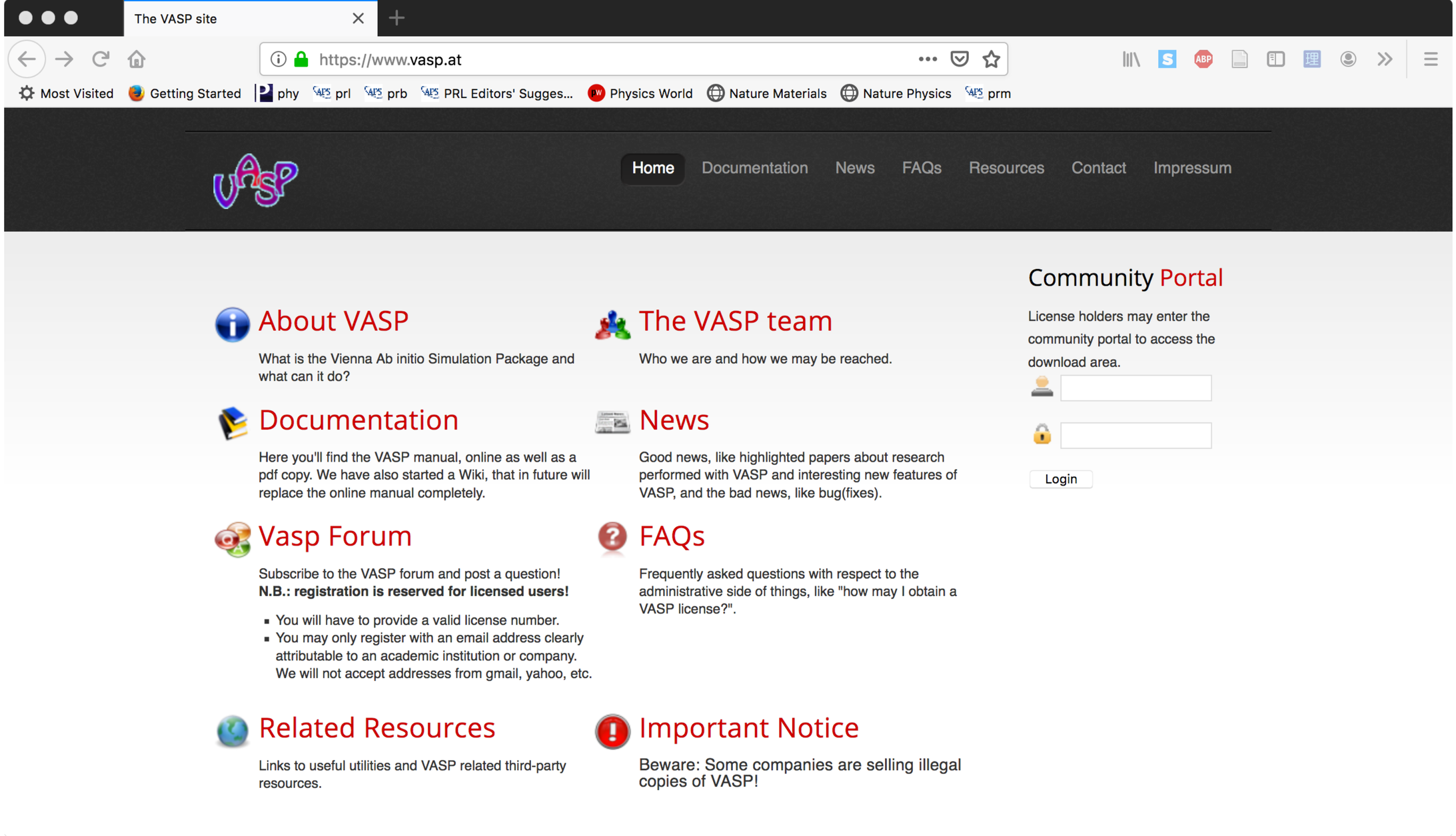
- **Kebnekaise**, HPC2N, UmU  this course
- Tetralith / Sigma, NSC, LiU
- Beskow, PDC, KTH
- Also available at other systems/centers

- On many systems:

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

VASP versions & utilities

- **Latest:** patch.5.4.4.16052018 for 18Apr17
  - Check centre 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



https://www.vasp.at/

vasp

https://cms.mpi.univie.ac.at/vasp/vasp/vasp.html

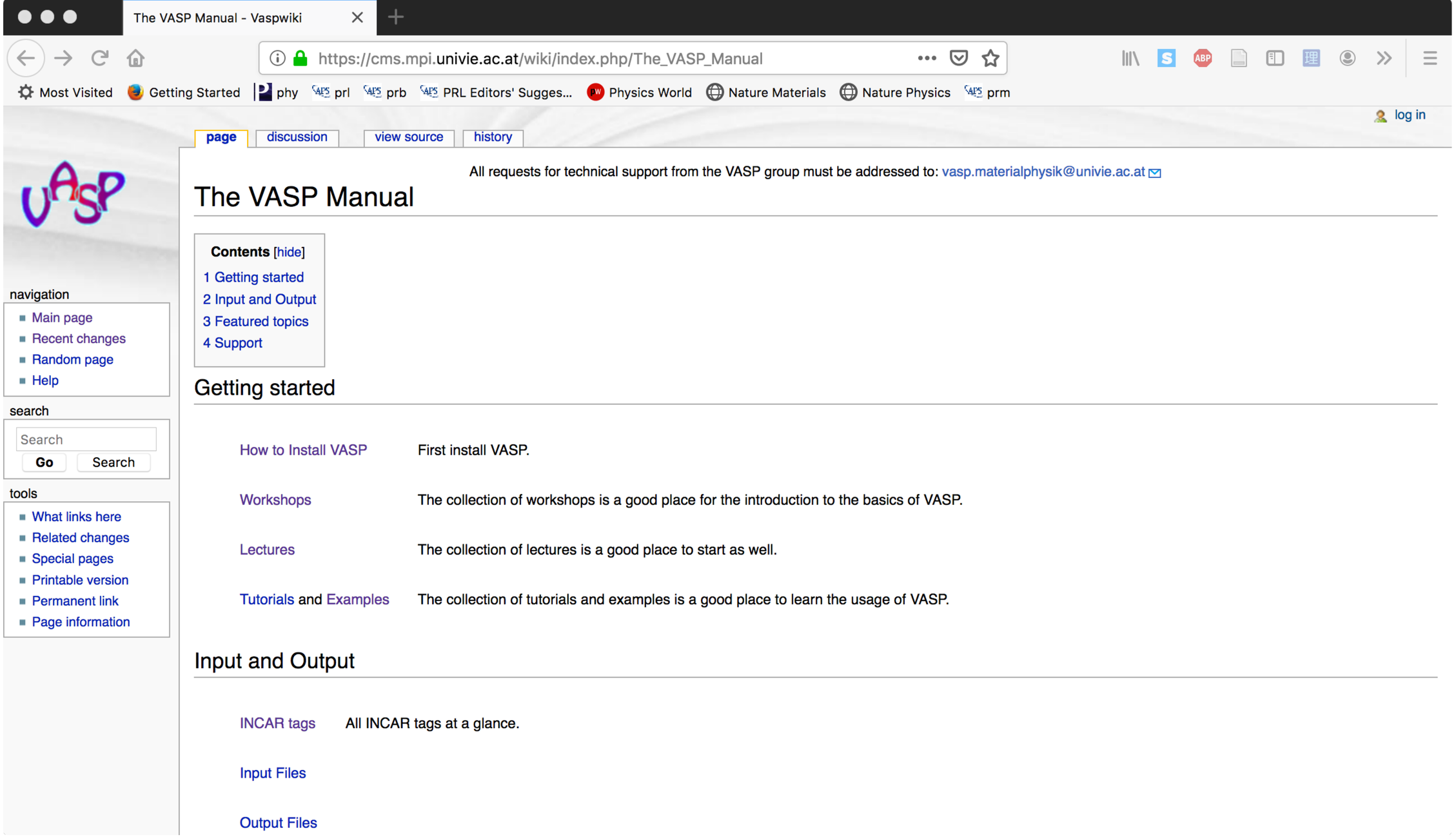
Most Visited | Getting Started | phy | prl | prb | PRL Editors' Sugges... | Physics World | Nature Materials | Nature Physics | prm

Next Up Previous Contents Index

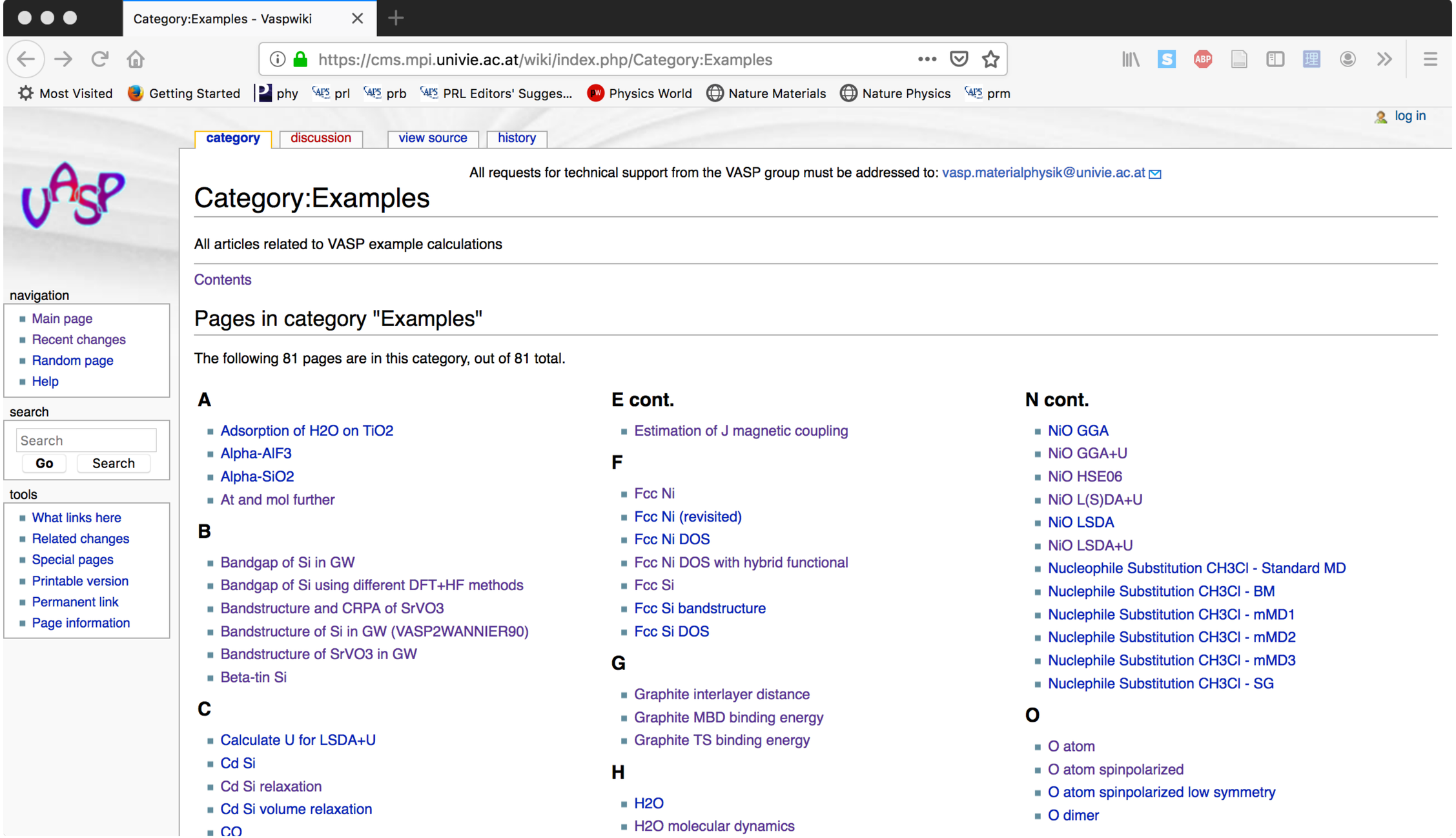
Next: Introduction **Contents** **Index**

N.B. This document is no longer maintained, please visit our wiki.

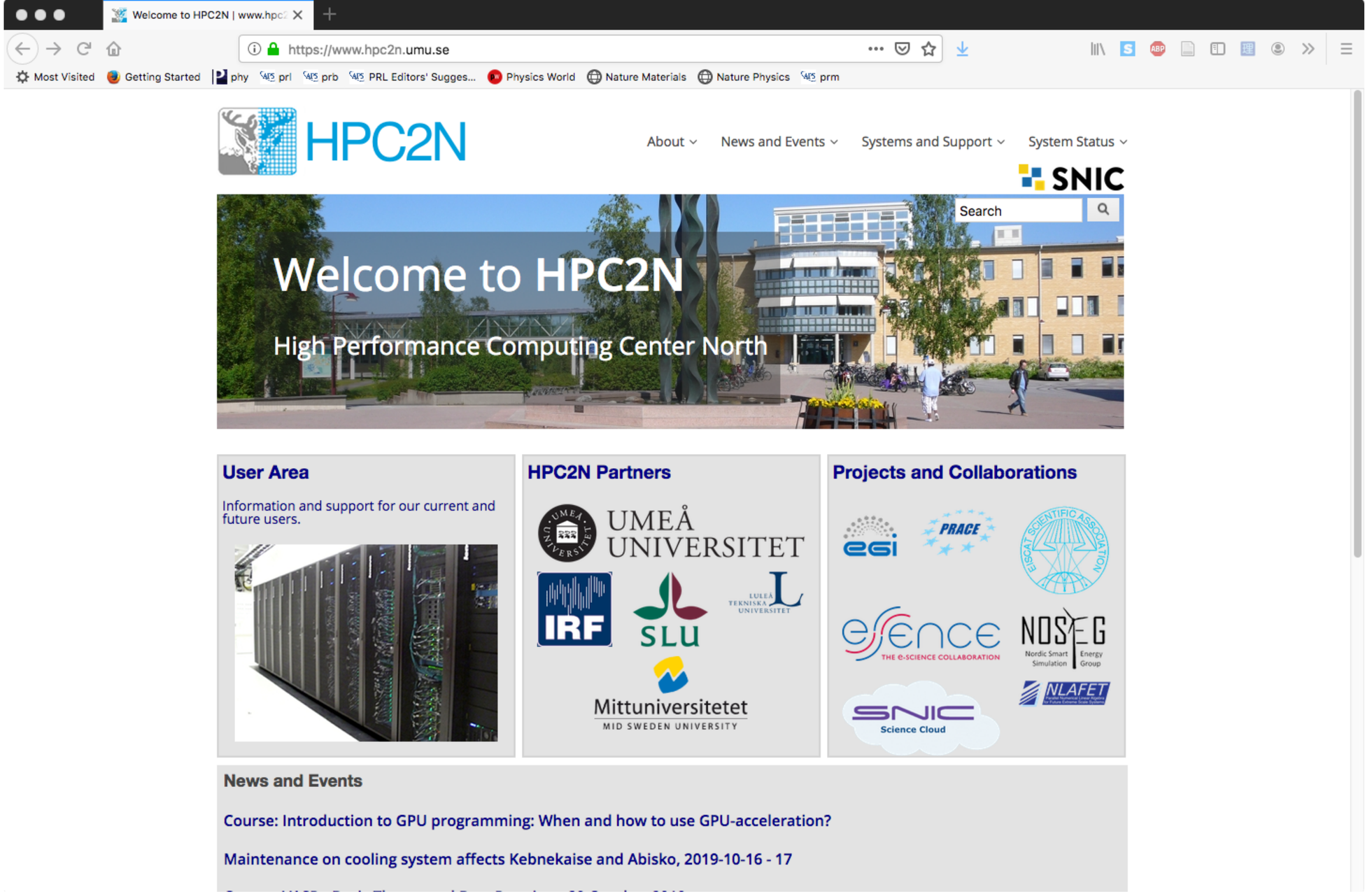
https://cms.mpi.univie.ac.at/vasp/vasp/vasp.html



https://cms.mpi.univie.ac.at/wiki/index.php/The_VASP_Manual



https://cms.mpi.univie.ac.at/wiki/index.php/The_VASP_Manual



https://www.hpc2n.umu.se/ Systems and Support > Software > VASP

https://www.hpc2n.umu.se/resources/software/vasp

Welcome to HPC2N | www.hpc2

https://www.hpc2n.umu.se

Most Visited | Getting Started | phy | prl | prb | PRL Editors' Sugges... | Physics World | Nature Materials | Nature Physics | prm

HPC2N

About | News and Events | Systems and Support | System Status

SNIC

Search

Welcome to HPC2N

High Performance Computing Center North

User Area

Information and support for our current and future users.

HPC2N Partners

UMEÅ UNIVERSITET

IRF

SLU

LULEÅ TEKNISKA UNIVERSITET

Mittuniversitetet
MID SWEDEN UNIVERSITY

Projects and Collaborations

EGI

PRACE

EISCAT SCIENTIFIC ASSOCIATION

eSCIENCE
THE e-SCIENCE COLLABORATION

NOSTEG
Nordic Smart Simulation Energy Group

NLAFET

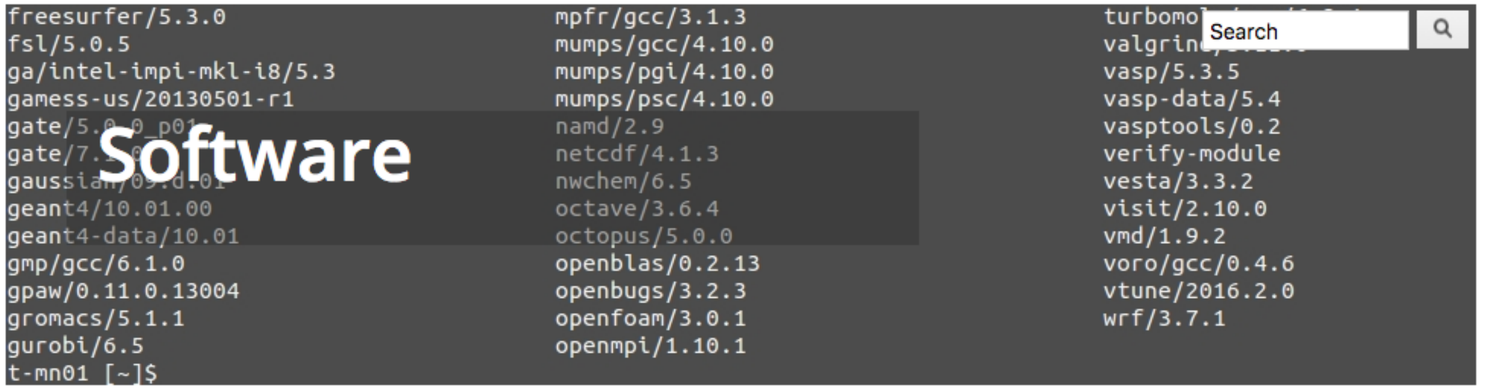
SNIC
Science Cloud

News and Events

Course: Introduction to GPU programming: When and how to use GPU-acceleration?

Maintenance on cooling system affects Kebnekaise and Abisko, 2019-10-16 - 17

	models of macromolecular complexes using Electron Microscopy. External info:	
Siesta	Siesta performs electronic structure calculations and ab initio molecular dynamics simulations of molecules and solids. External info:	Abisko, Kebnekaise
Turbomole	TURBOMOLE is a quantum chemical program package Extern info:	Abisko, Kebnekaise
VASP	Performs ab-initio quantum-mechanical molecular dynamics (MD) using pseudopotentials and a plane wave basis set. Extern info:	Abisko, Kebnekaise
VMD	VMD is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting. Extern info:	Abisko, Kebnekaise
WRF	Advanced Research WRF (ARW) Modeling System is a flexible, state-of-the-art atmospheric simulation system. Extern info:	Abisko, Kebnekaise
Compilers		
GCC	GNU Compiler Collection. Has C, C++, Fortran 77, Fortran 90, and Fortran 95. Available for serial, OpenMP, and MPI code for all languages, with the exception of Fortran 95 MPI programs. Part of several compiler toolchains. External info:	Abisko, Kebnekaise
Intel	Intel Compilers. Has C, C++, Fortran 77, Fortran 90, and Fortran 95. Available for serial, OpenMP, and MPI code for all	Abisko, Kebnekaise



VASP

Policy

The Vasp program is not distributed via site licences. However, HPC2N have access to the VASP code to be able to support any research groups that have a valid VASP license.

See the VASP license for information regarding terms for published work.

When you have gotten access to a license, the **license holder** should either add the license info into SUPR (or contact support@hpc2n.umu.se with the following information: license number and list of users who should have access). You will then be given access to using VASP.

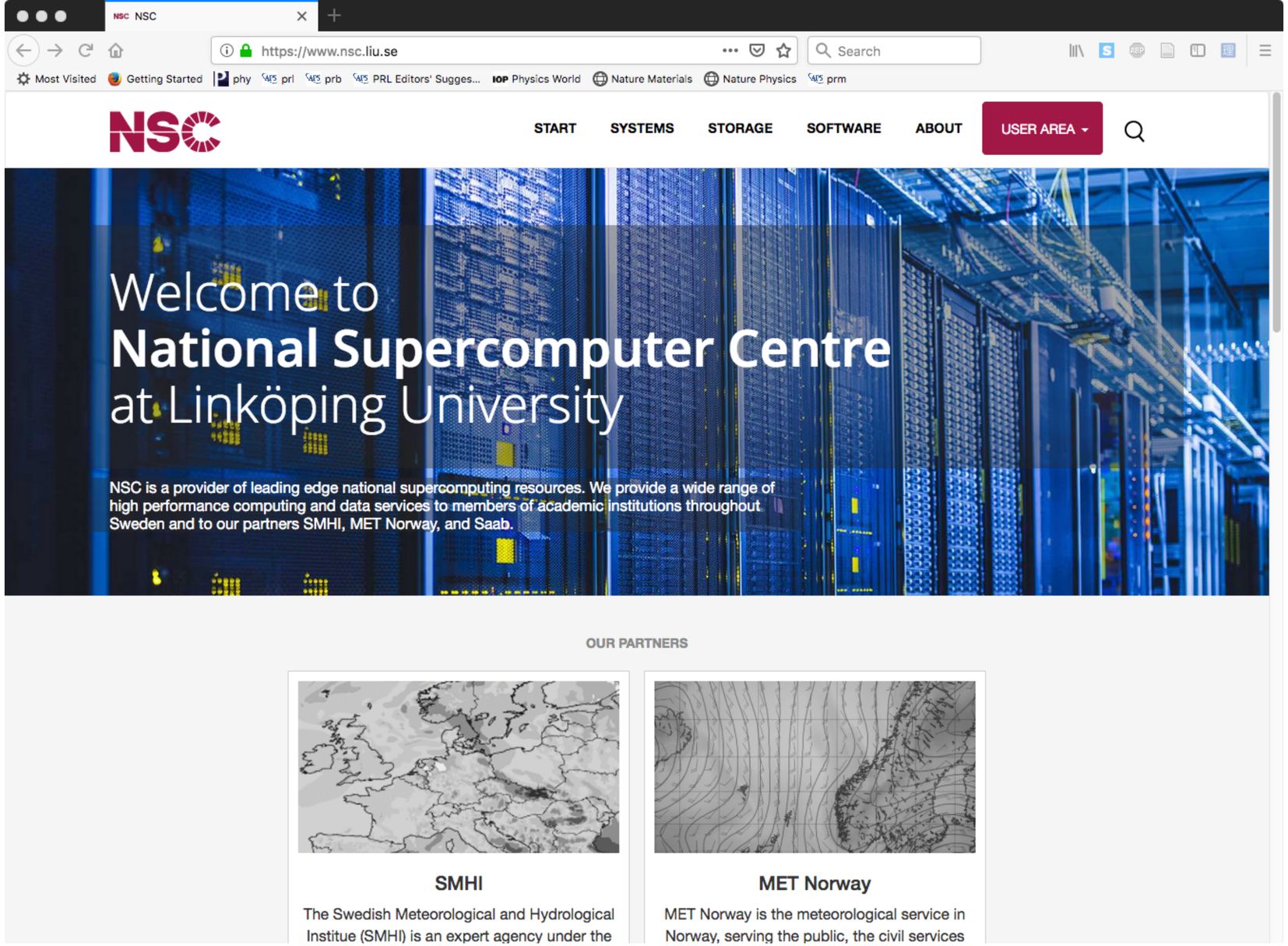
Note: only the owner of the license can add/delete users to/from the access list.

General

VASP is a package for performing ab-initio quantum-mechanical molecular dynamics (MD) using pseudopotentials and a plane wave basis set.

Description

VASP is a complex package for performing ab-initio quantum-mechanical molecular dynamics (MD) simulations using pseudopotentials or the



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

https://www.nsc.liu.se/software/installed/tetralith/vasp/

NSC NSC

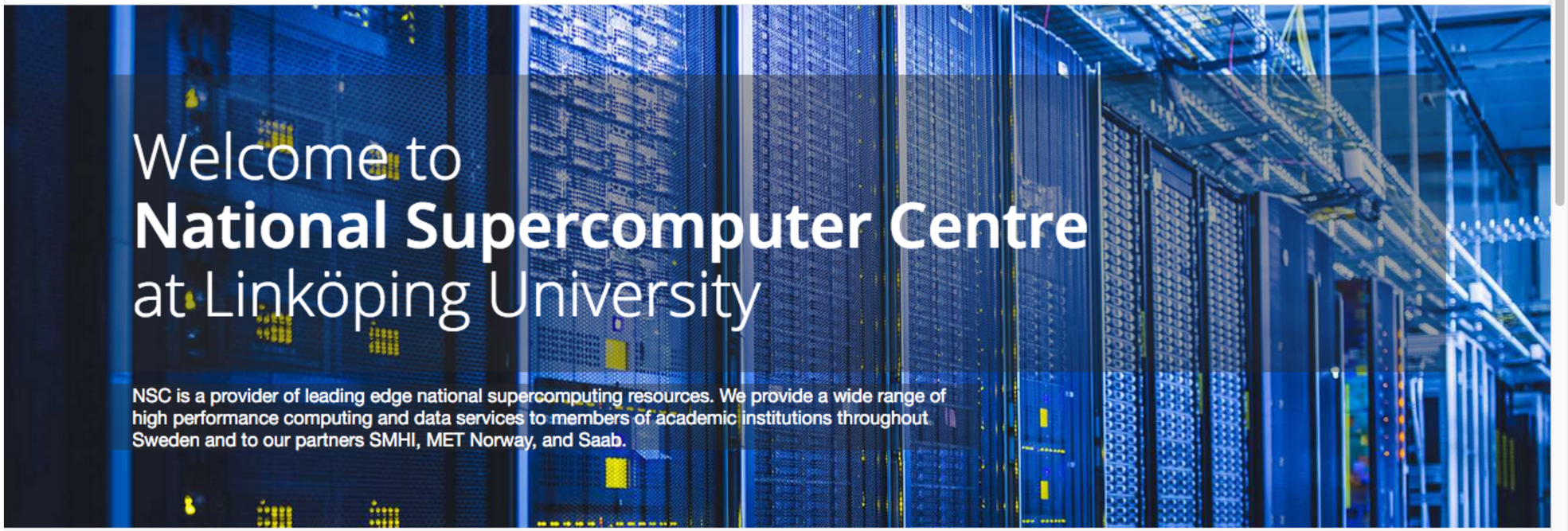
https://www.nsc.liu.se

Search

Most Visited | Getting Started | phy | prl | prb | PRL Editors' Sugges... | Physics World | Nature Materials | Nature Physics | prm

NSC

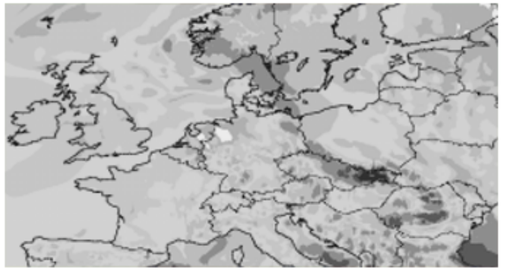
START SYSTEMS STORAGE SOFTWARE ABOUT USER AREA



Welcome to National Supercomputer Centre at Linköping University

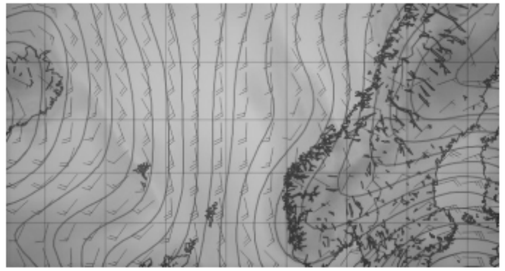
NSC is a provider of leading edge national supercomputing resources. We provide a wide range of high performance computing and data services to members of academic institutions throughout Sweden and to our partners SMHI, MET Norway, and Saab.

OUR PARTNERS



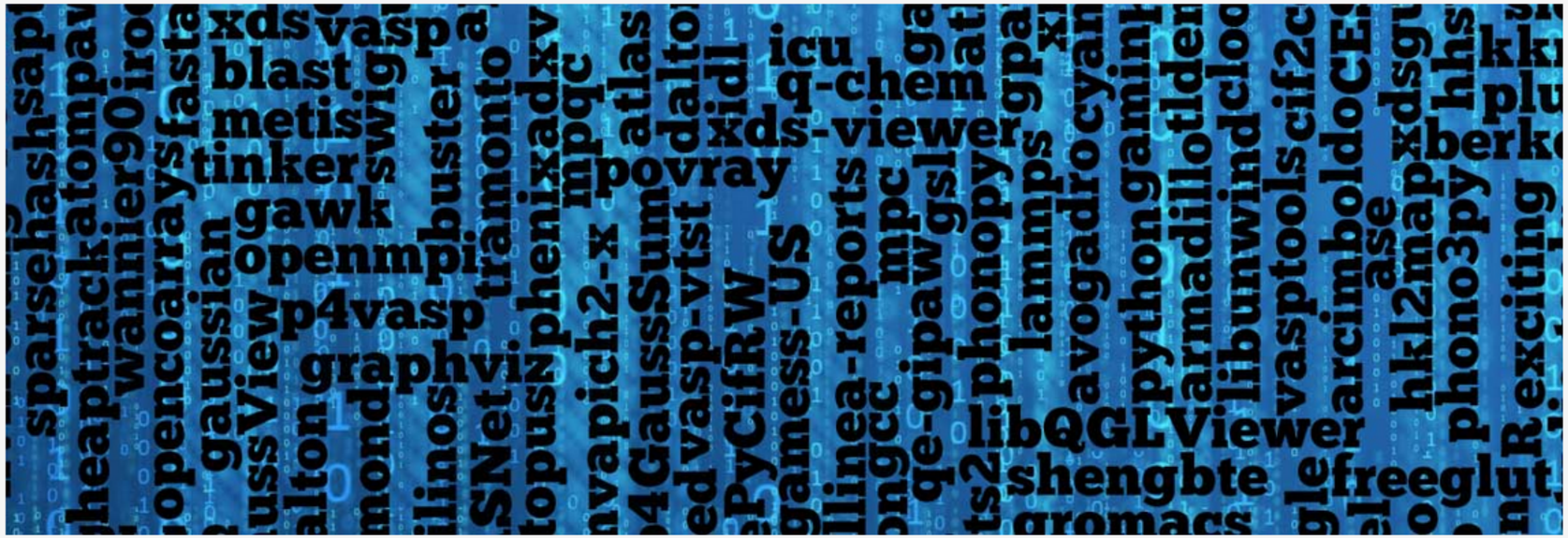
SMHI

The Swedish Meteorological and Hydrological Institue (SMHI) is an expert agency under the



MET Norway

MET Norway is the meteorological service in Norway, serving the public, the civil services



Software

Explore this part to find out about all software environment related matters. Please proceed to the installed software page to find out which scientific software we have on our clusters. If you are going to compile software from source code, we suggest that you read the NSC build environment introduction and the compilers section.

Installed software

Which software is available on what systems and how to run it

Software installation policy

What to do if software you need isn't installed

Software licensing

How we handle software licensing of commercial software.

Installed software

NSC has a large number of software installations available, often in multiple versions to suit the needs of various user communities. For a list of installed software, please see the corresponding resource page below. If you need software that is presently not installed, please see our software installation policy.

Software portfolios by cluster

- Tetralith & Sigma Software List.
- Bi Software List. This is the main resource for finding out what software is available on Bi.

Module system

You can also query the module system for available software and recommendations on what versions to use, e.g:

```
module avail
module add vasp/recommendation
```

SNIC knowledge base

Information on software and availability for all of SNIC is also available in the SNIC knowledge base software section. There is specific information for these NSC resources:

Tetralith & Sigma Software

A list of software installed on Tetralith and Sigma

Tetralith & Sigma Software

Tetralith & Sigma Software List

DISCLAIMER: Please note that the software catalogue is a work in progress! If your application is missing, please request it by sending e-mail to NSC Support

The following scientific applications have been installed centrally under `/software/sse/` . This list may not always be 100% up to date. The most reliable source is running the command `module avail` while logged into Tetralith or Sigma, possibly augmented by `ls /software/sse/manual/` to show additional manually performed installations without modules. Please note that some of this software is licensed, and may not be available for everyone. You need ask NSC for access, which is typically granted upon some proof of having a license.

The list was last updated: 2019-02-21

Electronic structure

- Abinit
- ASE
- Elk
- EPW
- exciting
- GPAW
- phonopy
- phono3py
- p4vasp
- Quantum Espresso
- vasptools
- VASP (licensed)
- WIEN2k (licensed)

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
#SBATCH -A SNIC-xxx-yyy

module add VASP/5.4.4.16052018-nsc1-intel-2018b-eb
mpprun vasp_[std/gam/ncl]
```

This script allocates 4 compute nodes with 32 cores each, for a total of 128 cores (or MPI ranks) and runs VASP in parallel using MPI. Note that you should edit the jobname and the account number before submitting.

Software

General information about VASP

Licenses

General information about VASP

System	Available versions
Beskow	5.4.4-wannier90, 5.4.4, 5.4.1.patched, 5.3.5-vtst3.1, 5.3.5-31Mar14
Tegner	5.4.1, 5.3.5

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: http://vasp.at

Licenses

VASP is not free software and requires a software license. VASP licenses are managed in SUPR. All people who want to use VASP should have SUPR accounts and be a member of a VASP group in SUPR. VASP groups have owners, typically a principal investigator of a project, and that owner can add and remove people using the SUPR interface. If you are Ph.D student, we suggest that you check with your supervisor.

Using VASP 5.4.4 on Beskow

General observations

- VASP is not helped by hyper-threading (64 virtual cores per compute node).
- No GPU/OpenMP-support.
- Running on fewer than 32 cores per node allocates more memory to each MPI task. This can in some cases improve performance and is necessary if your job crashes with an OOM error. See the example submit script below on how to do this correctly.

NPAR, NCORE and NSIM

From initial testing, we recommend:

- NPAR = number of compute nodes
- NCORE = cores / node, typically 16,24 or 32.
- NSIM = 2
- KPAR = number of compute nodes (if applicable)

How to choose the number of cores

Rule of thumb:

- 1 atom per core = Good
- 0.5 atom per core = Could work (but bad efficiency and time wasted)

Example of day-to-day tools

- less / vi — reading/editing files
- grace / gnuplot — plotting tools
- Bash — simple scripts
- cif2cell — convert from .cif
- p4vasp — analysis of VASP output
- xcrysden / vesta — view structure
- Schrödinger — create/view structure

@Kebnekaise, Tetralith

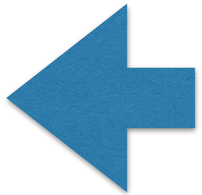
Also of interest:

- Python / R — analysis etc. (check out e.g. using jupyter)
- ASE — different tools and Python modules
- Matlab / Octave — analysis etc.

Check out presentations for Tetralith training

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
- Used within a special framework (VTST, …)
- See VASP wiki examples



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 SNIC 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 the settings in the 2nd part

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…

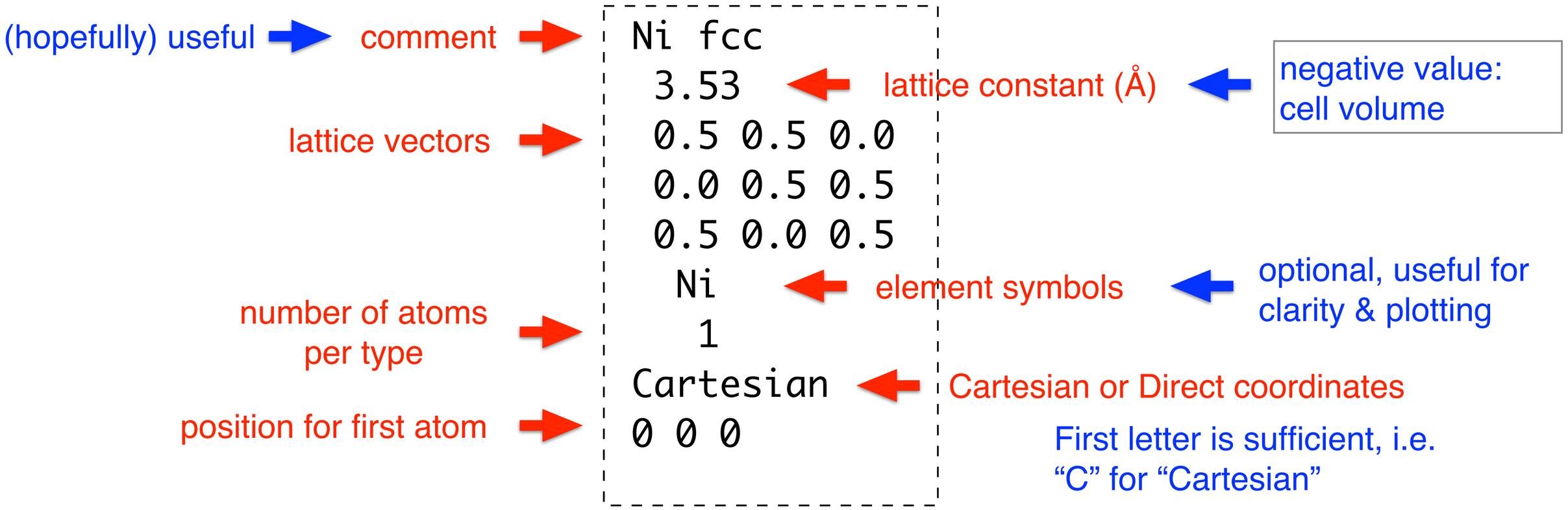
INCAR defaults

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 the course 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

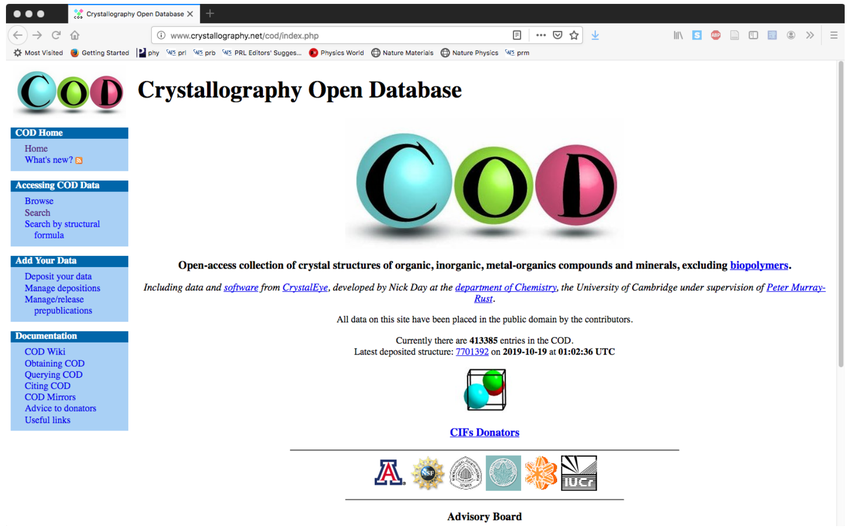
Rest of H, Si & Ag atoms following

POSCAR

Some useful resources:

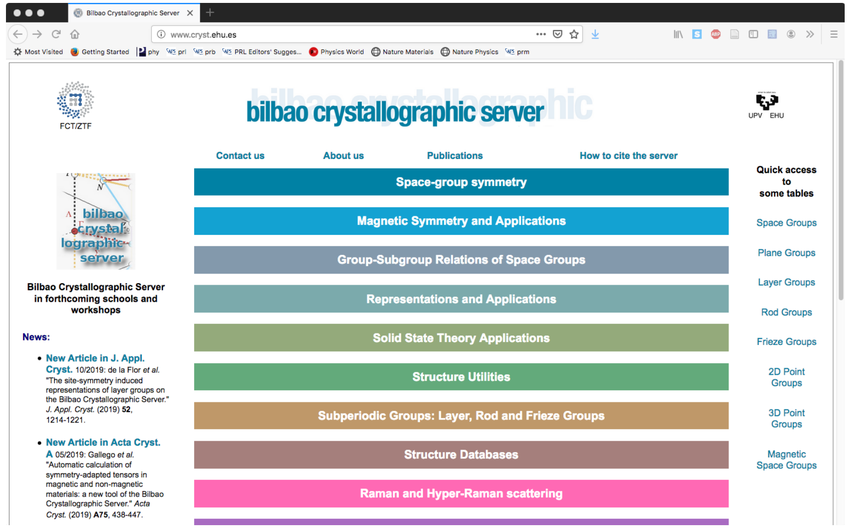
[Crystallography Open Database](#)

Database with published structures from experiment `.cif`



[Bilbao Crystallographic Server](#)

Crystallographic tools



POSCAR

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

[Atomic Simulation Environment (ASE)](#)

Handle structures (and much more) using python scripts

[cif2cell](#)

Versatile script, reads `.cif` saves to many formats including POSCAR

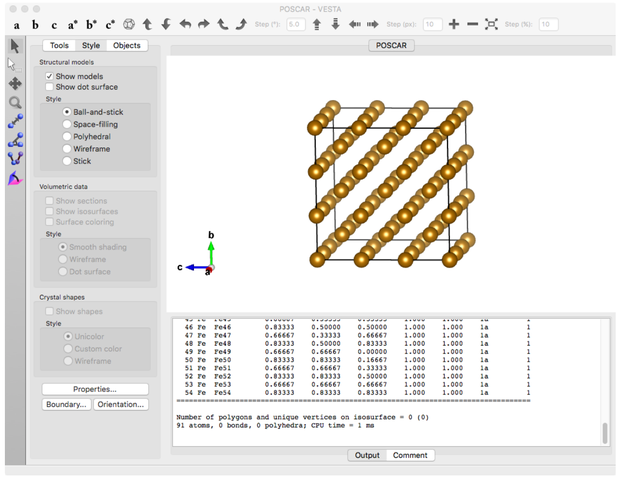
Commercial software:

[NanoLab](#)

[MedeA](#)

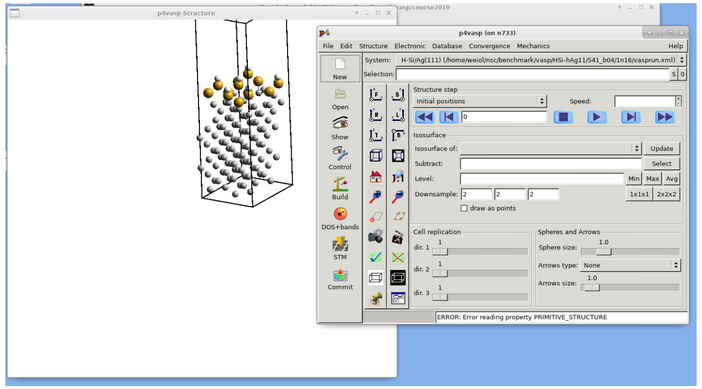
[MaterialsStudio](#)

[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
  11.0000000000000
 parameters from PSCTR are:
   VRHFIN =Cu: d10 p1
   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)
   ENMAX  =  295.446; ENMIN  =  221.585 eV
   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
```

type, element, date

valence

atomic configuration

XC-type

energy cutoff

smallest energy cutoff

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
```

@Kebnekaise:
```
$ echo $VASP_PP_PATH
/hpc2n/eb/software/Core/VASP-data/5.4/potPP
```

@Tetralith:
```
/software/sse/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

Course 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

real vs. reciprocal space

- 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

VASP binaries

- vasp_std - regular version
- vasp_gam - one k-point (Gamma), faster
- vasp_ncl - noncollinear magnetism
- vasp_gpu, vasp_gpu_ncl - for GPU
- + modifications

e.g. constrained relaxation

Job script - Kebnekaise (HPC2N)

```
#!/bin/bash
#SBATCH -A snic2019-3-203
#SBATCH -J test
#SBATCH -t 3:59:00
#SBATCH -n 56

ml icc/2017.4.196-GCC-6.4.0-2.28
ml ifort/2017.4.196-GCC-6.4.0-2.28
ml impi/2017.3.196
ml VASP/5.4.4-18Apr17-hpc2n

mpirun vasp_std
```

```
#SBATCH -N 2
```

Example: running on 2 nodes (28x2 cores) @Kebnekaise

https://www.hpc2n.umu.se/resources/software/vasp

running @Abisko: note differences

Job script - Kebnekaise (HPC2N)

```
#!/bin/bash
#SBATCH -A snic2019-3-203
#SBATCH -J test
#SBATCH -t 3:59:00
#SBATCH -N 2
#SBATCH --ntasks-per-node=14

ml icc/2017.4.196-GCC-6.4.0-2.28
ml ifort/2017.4.196-GCC-6.4.0-2.28
ml impi/2017.3.196
ml VASP/5.4.4-18Apr17-hpc2n

mpirun vasp_std
```

```
#SBATCH -n 56
```

Example: running on 2 nodes (28x2 cores), using half the cores for more memory @Kebnekaise

https://www.hpc2n.umu.se/resources/software/vasp

running @Abisko: note differences

Job script - Kebnekaise (HPC2N)

GPU calc.

```
#!/bin/bash
#SBATCH -A snic2019-3-203
#SBATCH -J test
#SBATCH -t 3:59:00
#SBATCH -n 28
#SBATCH --gres=gpu:v100:2,gpuexcl

ml icc/2017.4.196-GCC-6.4.0-2.28
ml ifort/2017.4.196-GCC-6.4.0-2.28
ml impi/2017.3.196
ml CUDA/9.1.85
ml VASP/5.4.4-18Apr17-p01-hpc2n

mpirun vasp_gpu
```

Example: running on 1 node (28 cores) with 2xV100 GPUs @Kebnekaise

https://www.hpc2n.umu.se/resources/software/vasp

Job script - Tetralith (NSC)

```
#!/bin/bash
#SBATCH -A snic2019-3-203
#SBATCH -J test
#SBATCH -t 3:59:00
#SBATCH -N 2

module load VASP/5.4.4.16052018-nsc1-intel-2018a-eb
mpprun vasp
```

`#SBATCH -n 64`

note “mpprun”

Example: running on 2 nodes (32x2 cores) @Tetralith

To increase available memory, reduce cores/node, e.g:
`#SBATCH --ntasks-per-node=16`

Alternatively, use “fat” memory nodes:
`#SBATCH -C fat`

https://www.nsc.liu.se/software/installed/tetralith/vasp/

Job script - Beskow (PDC)

```
#!/bin/bash -l
#SBATCH -A 2019-3-203
#SBATCH -J test
#SBATCH -t 3:59:00
#SBATCH --nodes=2

module unload cray-mpich/7.0.4
module load vasp/5.4.4

aprun -n 48 -N 24 vasp
#aprun -n 64 -N 32 vasp
```

Example: running on 2 nodes (24x2 cores) @Beskow
alternatively on 2 nodes (32x2 cores)

https://www.pdc.kth.se/software/software/VASP/beskow/5.4.4/index.html#running-vasp

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

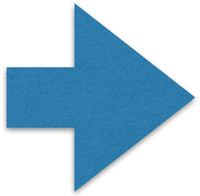
Charge density residual vector

	Step N	Total free Energy E	Energy diff. dE	Eigenvalue diff. 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.)             |
|                                                                             |
 -----------------------------------------------------------------------------
```