

2. VASP - Basics

Weine Olovsson

National Supercomputer Centre (NSC), Linköping University

SNIC-PRACE training, online @NSC 22-23rd Feb 2022

VASP - Best Practices Workshop



<https://www.nsc.liu.se/>



<https://www.snic.se/>



<https://training.prace-ri.eu/>



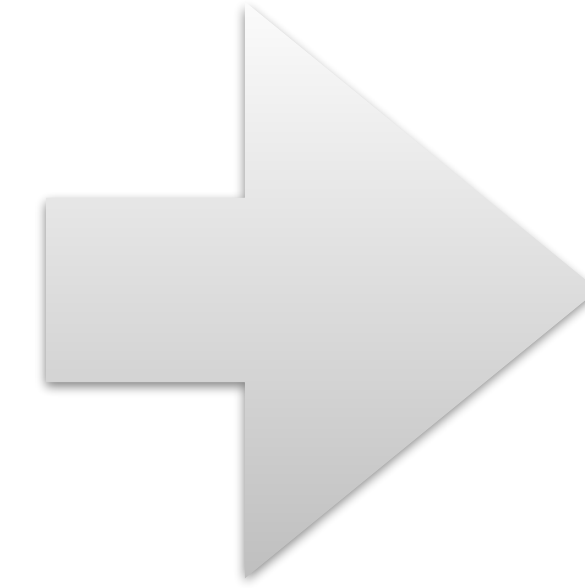
Introduction

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

... clickable links are [underlined](#)

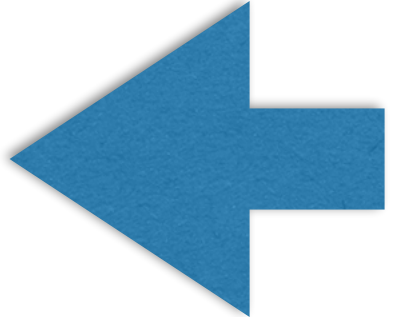
Short background

- 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
- **VASP6** is available since 2020



<https://vasp.at/>

Starting advice

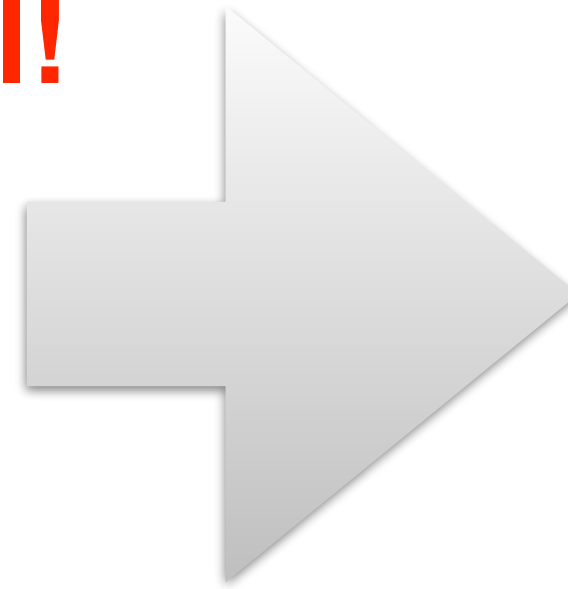
- Read the [documentation!](#)
- VASP default settings  good starting point
- Caution: “inherited” input files
- Avoid overly messy INCAR...
- Possible differences in installations & versions
refer to respective center webpages / documentation

Resources

- Wiki / Manual

Check in detail!

- Wiki
examples,
presentations
- Forum

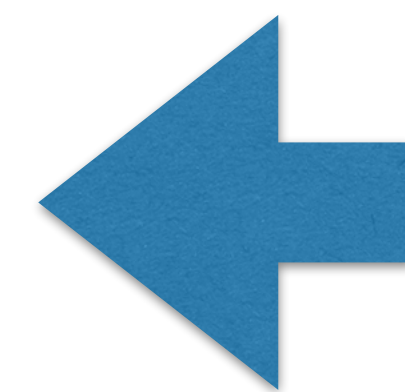


Find the links:

<https://vasp.at/>

- Peter Larsson's (old) blog at NSC:

<https://www.nsc.liu.se/~pla/>

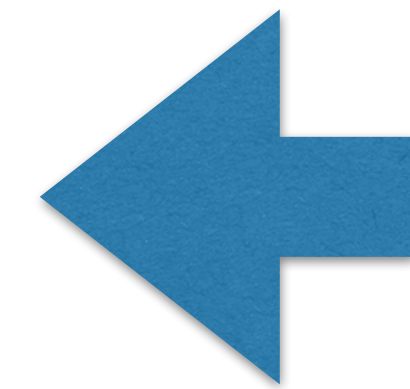


info &
discussion

Questions / trouble? support@nsc.liu.se, ...

VASP at SNIC HPC centers

- **Tetralith** / Sigma, NSC, LiU



this course

- Kebnekaise, HPC2N, UmU

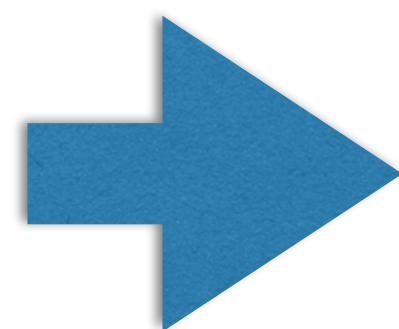
- Dardel, PDC, KTH

- Also available at other systems/centers

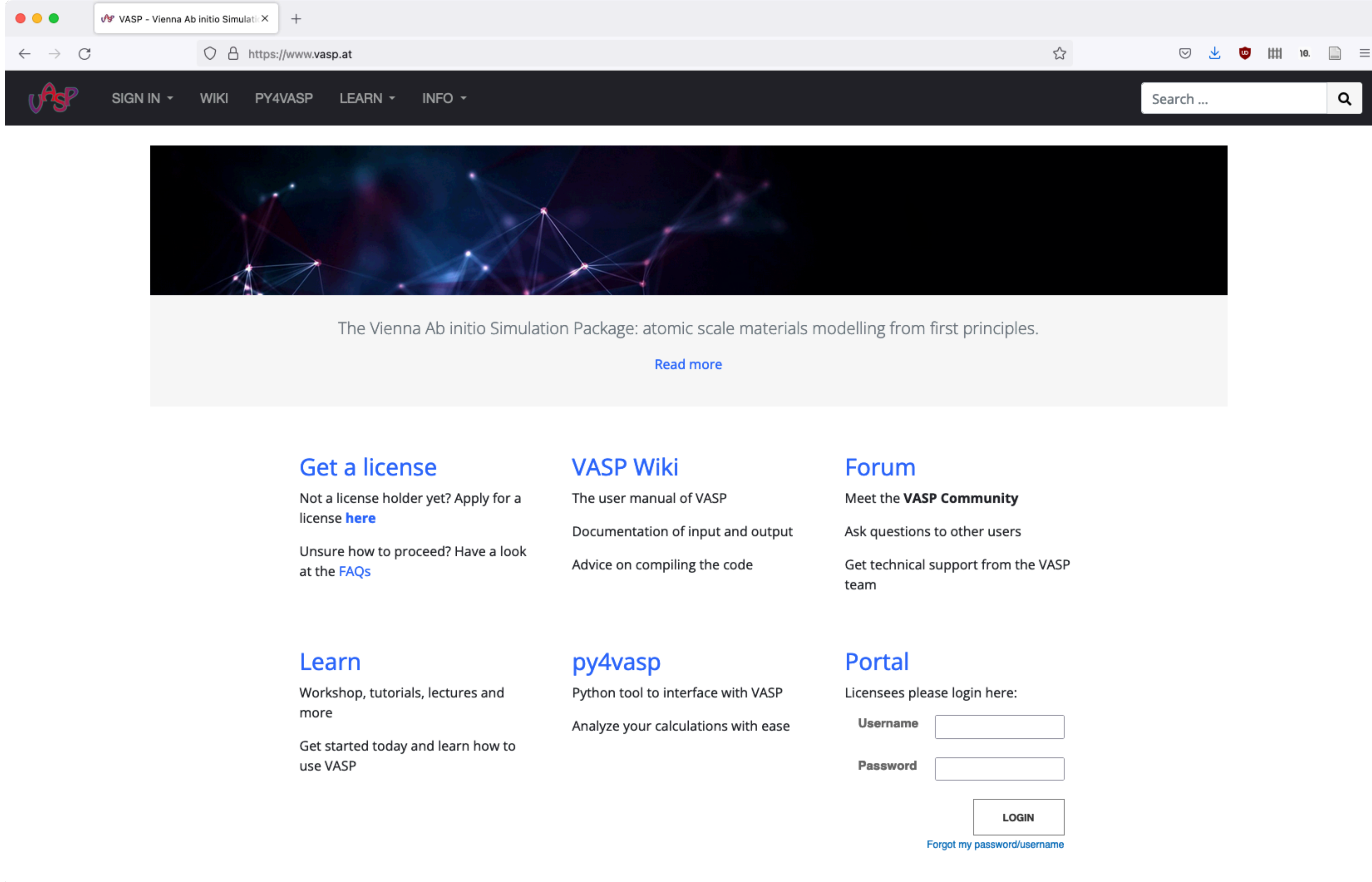
- **On many systems:** \$ module avail vasp
\$ module spider vasp

VASP versions & utilities

- **Latest:** 6.3.0 (from Jan -22)
- 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



...more about utilities tomorrow



<https://www.vasp.at/>

...was updated just before this workshop!

- py4vasp
- Installation
- Quick start
- calculation
- raw
- data
- exceptions

py4vasp

py4vasp is a python interface to extract data from VASP calculations. It is intended mainly to get a quick look at the data and provide the functionality to export it into common formats that can be used by other more sophisticated postprocessing tools. The second domain of application is for people that want to write python scripts based on the data calculated by VASP. This tool interfaces directly with the new HDF5 file format and thereby avoids parsing issues associated with the XML or OUTCAR files.

For these two groups of users, we provide a different level of access. The simple routines used in the tutorials will read the data from the file directly and then generate the requested plot. For script developers, we provide an expert interface where the data is lazily loaded as needed with some greater flexibility when the data file is opened and closed.

Installation

While this is not required to be able to run *py4vasp*, you may want to consider creating a separate environment for installation to avoid interference with other installed packages. ¹ You can then install *py4vasp* from [PyPI](#) using the pip package installer

```
pip install py4vasp
```

This will automatically download *py4vasp* as well as all the required dependencies. However, we noticed that this approach is not fail-safe, because the installation of the *mdtraj* dependency does not work on all operating systems. So in case the simple installation above fails, you may need to use *conda* to install *mdtraj*

```
conda install -c conda-forge mdtraj  
pip install py4vasp
```

If these commands succeed, you should be able to use *py4vasp*. You can make a quick test of your installation running the following command

```
python -c "import py4vasp; print(py4vasp.__version__)"
```

This should print the version of *py4vasp* that you installed.

Important



The Vienna Ab initio Simulation Package: atomic scale materials modelling from first principles.

[Read more](#)

Get a license

Not a license holder yet? Apply for a license [here](#)

Unsure how to proceed? Have a look at the [FAQs](#)



VASP Wiki

The user manual of VASP

Documentation of input and output

Advice on compiling the code



Forum

Meet the **VASP Community**

Ask questions to other users

Get technical support from the VASP team

Learn

Workshop, tutorials, lectures and more

Get started today and learn how to use VASP

py4vasp

Python tool to interface with VASP

Analyze your calculations with ease

Portal

Licensees please login here:

Username

Password

LOGIN

[Forgot my password/username](#)

The VASP Manual - Vaspwiki

https://www.vasp.at/wiki/index.php/The_VASP_Manual

Log in

Page **Discussion** Read **View source** **View history** Search Vaspwiki

Requests for technical support from the VASP group should be posted in the [VASP-forum](#).

The VASP Manual

(Redirected from [Main page](#))

Contents [hide]

- [1 Getting started](#)
- [2 Input and Output](#)
- [3 Featured topics](#)
- [4 Support](#)

Getting started

| | |
|--|---|
| VASP6 | Features that will only be available in VASP.6.X. |
| How to Install VASP | First install VASP. |
| Workshops | The collection of workshops is a good place for the introduction to the basics of VASP. |
| Lectures | The collection of lectures is a good place to start as well. |
| Tutorials and Examples | The collection of tutorials and examples is a good place to learn the usage of VASP. |

Input and Output

| | |
|------------------------------|-----------------------------|
| INCAR tags | All INCAR tags at a glance. |
| Input Files | |
| Output Files | |

Featured topics

| Category | subtopics (amongst others) |
|----------|----------------------------|
|----------|----------------------------|



- [Main page](#)
 - [Recent changes](#)
 - [Random page](#)
 - [Help about MediaWiki](#)
- Tools
- [What links here](#)
 - [Related changes](#)
 - [Special pages](#)
 - [Printable version](#)
 - [Permanent link](#)
 - [Page information](#)

https://www.vasp.at/wiki/index.php/The_VASP_Manual

The VASP Manual - Vaspwiki

https://www.vasp.at/wiki/index.php/The_VASP_Manual

Featured topics

| Category | <i>subtopics (amongst others)</i> |
|---|--|
| PP and PAW | Theory, available potentials. |
| Electronic Minimization | Davidson, RMM-DIIS, Conjugate-gradient, preconditioning, mixing. |
| XC Functionals | LDA, GGA, metaGGA, HF/DFT hybrid functionals, vdW functionals. |
| Structural Optimization | Ionic minimization methods. |
| Molecular Dynamics | Barostats, thermostats, ensembles, different MD methods. |
| Machine Learning | Machine-learning force fields. |
| Lattice Vibrations | Phonons, electron-phonon interactions. |
| Dielectric Properties | Static and frequency dependent dielectric properties, Berry phases, spectroscopy (UV, VIS, X-ray). |
| Linear Response | Static dielectric properties, phonons form linear response. |
| NMR | Chemical shifts, electric field gradients. |
| Many-Body Perturbation Theory | ACFDT, BSE, GW, MP2, CRPA. |
| Magnetism | Spin-orbit coupling, non-collinear magnetism. |
| Defects | Dipole corrections for defects in solids. |
| Atoms and Molecules | Monopole, dipole, and quadrupole corrections. |
| Transition States | Elastic band method, improved dimer method. |
| Wannier Functions | Obtaining Wannier functions. |
| Common Pitfalls | Most important parameters, typical sources of errors. |
| Theoretical Background | DFT, PAW, vdW, spectroscopy, MD. |
| VASP.6 / VASP6 | Features that will only be available in VASP.6.X. |

https://www.vasp.at/wiki/index.php/The_VASP_Manual

Category:Examples - Vaspwiki

Category:Examples

Log in

Category **Discussion** Read View source View history Search Vaspwiki

Requests for technical support from the VASP group should be posted in the VASP-forum.

Category:Examples

All articles related to VASP example calculations

Contents

Pages in category "Examples"


The following 80 pages are in this category, out of 80 total.

| | | |
|---|---|--|
| <p>A</p> <ul style="list-style-type: none"> • Adsorption of H2O on TiO2 • Alpha-AlF3 • Alpha-SiO2 • At and mol further <p>B</p> <ul style="list-style-type: none"> • Band gap renormalization in diamond using one-shot method • Bandgap of Si in GW • Bandgap of Si using different DFT+HF methods • Bandstructure and CRPA of SrVO3 • Bandstructure of Si in GW (VASP2WANNIER90) • Bandstructure of SrVO3 in GW • Beta-tin Si <p>C</p> <ul style="list-style-type: none"> • Calculate U for LSDA+U • Cd Si • Cd Si relaxation • Cd Si volume relaxation • CO • CO on Ni 111 surface • CO partial DOS • CO vibration • Collective jumps of a Pt adatom on fcc-Pt (001): Nudged Elastic Band Calculation | <ul style="list-style-type: none"> • Estimation of J magnetic coupling <p>F</p> <ul style="list-style-type: none"> • Fcc Ni • Fcc Ni (revisited) • Fcc Ni DOS • Fcc Ni DOS with hybrid functional • Fcc Si • Fcc Si bandstructure • Fcc Si DOS <p>G</p> <ul style="list-style-type: none"> • Graphite interlayer distance • Graphite MBD binding energy • Graphite TS binding energy <p>H</p> <ul style="list-style-type: none"> • H2O • H2O molecular dynamics • H2O vibration <p>I</p> <ul style="list-style-type: none"> • Improving the dielectric function • Including the Spin-Orbit Coupling • Ionic contributions to the frequency dependent dielectric function of NaCl | <ul style="list-style-type: none"> • Ni 111 surface relaxation • NiO • NiO GGA • NiO GGA+U • NiO HSE06 • NiO LSDA+U • Nucleophile Substitution CH3Cl - Standard MD • Nucleophile Substitution CH3Cl - BM • Nucleophile Substitution CH3Cl - mMD1 • Nucleophile Substitution CH3Cl - mMD2 • Nucleophile Substitution CH3Cl - mMD3 • Nucleophile Substitution CH3Cl - SG <p>O</p> <ul style="list-style-type: none"> • O atom • O atom spinpolarized • O atom spinpolarized low symmetry • O dimer <p>P</p> <ul style="list-style-type: none"> • Partial DOS of CO on Ni 111 surface • Plotting the BSE fatband structure of Si <p>S</p> <ul style="list-style-type: none"> • Si bandstructure • Si HSE bandstructure |
|---|---|--|


https://www.vasp.at/wiki/index.php/The_VASP_Manual

Welcome to HPC2N | www.hpc2n.umu.se


https://www.hpc2n.umu.se



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



Search




Welcome to HPC2N


High Performance Computing Center North

User Area

Information and support for our current and future users.




HPC2N Partners



New Director to lead HPC2N

Professor Paolo Bientinesi has been appointed as the new Director of The High Performance Computing Center North in Umeå, as professor Bo Kågström steps down.



News and Events

The SNIC Large and LUMI Sweden Spring 2022 calls are open. Deadline for submission is 1 April 2022 at 3 PM CEST.

New European collaboration for open science training

Course: An introduction to shared memory parallel programming using OpenMP, 28 February - 3 March, 2022

Who we are and what we do

High Performance Computing Center North (HPC2N) is a national center for Scientific and Parallel Computing.

<https://www.hpc2n.umu.se/>

Systems and Support > Software > VASP

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



User Area

Information and support for our current and future users.



HPC2N Partners



New Director to lead HPC2N

Professor Paolo Bientinesi has been appointed as the new Director of The High Performance Computing Center North in Umeå, as professor Bo Kågström steps down.



News and Events

The SNIC Large and LUMI Sweden Spring 2022 calls are open. Deadline for submission is 1 April 2022 at 3 PM CEST.

[New European collaboration for open science training](#)

[Course: An introduction to shared memory parallel programming using OpenMP, 28 February - 3 March, 2022](#)

Who we are and what we do

High Performance Computing Center North (HPC2N) is a national center for Scientific and Parallel Computing.

| | |
|--------------------|--|
| | External info: ↗ |
| Siesta | Siesta performs electronic structure calculations and ab initio molecular dynamics simulations of molecules and solids. External info: ↗ |
| Singularity | Singularity is a free, cross-platform and open-source computer program that performs operating-system-level virtualization also known as containerization. External info: ↗ |
| Turbomole | TURBOMOLE is a quantum chemical program package External info: ↗ |
| VASP | Performs ab-initio quantum-mechanical molecular dynamics (MD) using pseudopotentials and a plane wave basis set. External info: ↗ |
| VMD | VMD is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting. External info: ↗ |
| WRF | Advanced Research WRF (ARW) Modeling System is a flexible, state-of-the-art atmospheric simulation system. External info: ↗ |
| 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: ↗ |
| Intel | Intel Compilers. Has C, C++, Fortran 77, Fortran 90, and Fortran 95. Available for serial, OpenMP, and MPI code for all languages. Part of several compiler toolchains . External info: ↗ |
| PGI | Portland Group Compilers. 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 96 MPI programs. |





```
freesurfer/5.3.0      mpfr/gcc/3.1.3      turbomole/6.11.0
fsl/5.0.5            mumps/gcc/4.10.0   valgrind/3.12.0
ga/intel-impi-mkl-i8/5.3  mumps/pgi/4.10.0  vasp/5.3.5
gamess-us/20130501-r1  mumps/psc/4.10.0  vasp-data/5.4
gate/5.0.0_p01       namd/2.9           vasp-tools/0.2
gate/7.2.0           netcdf/4.1.3      verify-module
gaussian/09.d.01     nwchem/6.5        vesta/3.3.2
geant4/10.01.00      octave/3.6.4      visit/2.10.0
geant4-data/10.01   octopus/5.0.0     vmd/1.9.2
gmp/gcc/6.1.0       openblas/0.2.13   voro/gcc/0.4.6
gpaw/0.11.0.13004   openbugs/3.2.3    vtune/2016.2.0
gromacs/5.1.1       openfoam/3.0.1    wrf/3.7.1
gurobi/6.5          openmpi/1.10.1
t-mn01 [~]$
```

Software

[Home](#) » [VASP](#)

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 projector-augmented wave method and a plane wave basis set. The approach implemented in VASP is based on the (finite-temperature) local-density approximation with the free energy as variational quantity and an exact evaluation of the instantaneous electronic ground state at each MD time step.

VASP uses efficient matrix diagonalisation schemes and an efficient Pulay/Broyden charge density mixing. These techniques avoid all problems possibly occurring in the original Car-Parrinello method, which is based on the simultaneous integration of electronic and ionic equations of motion.

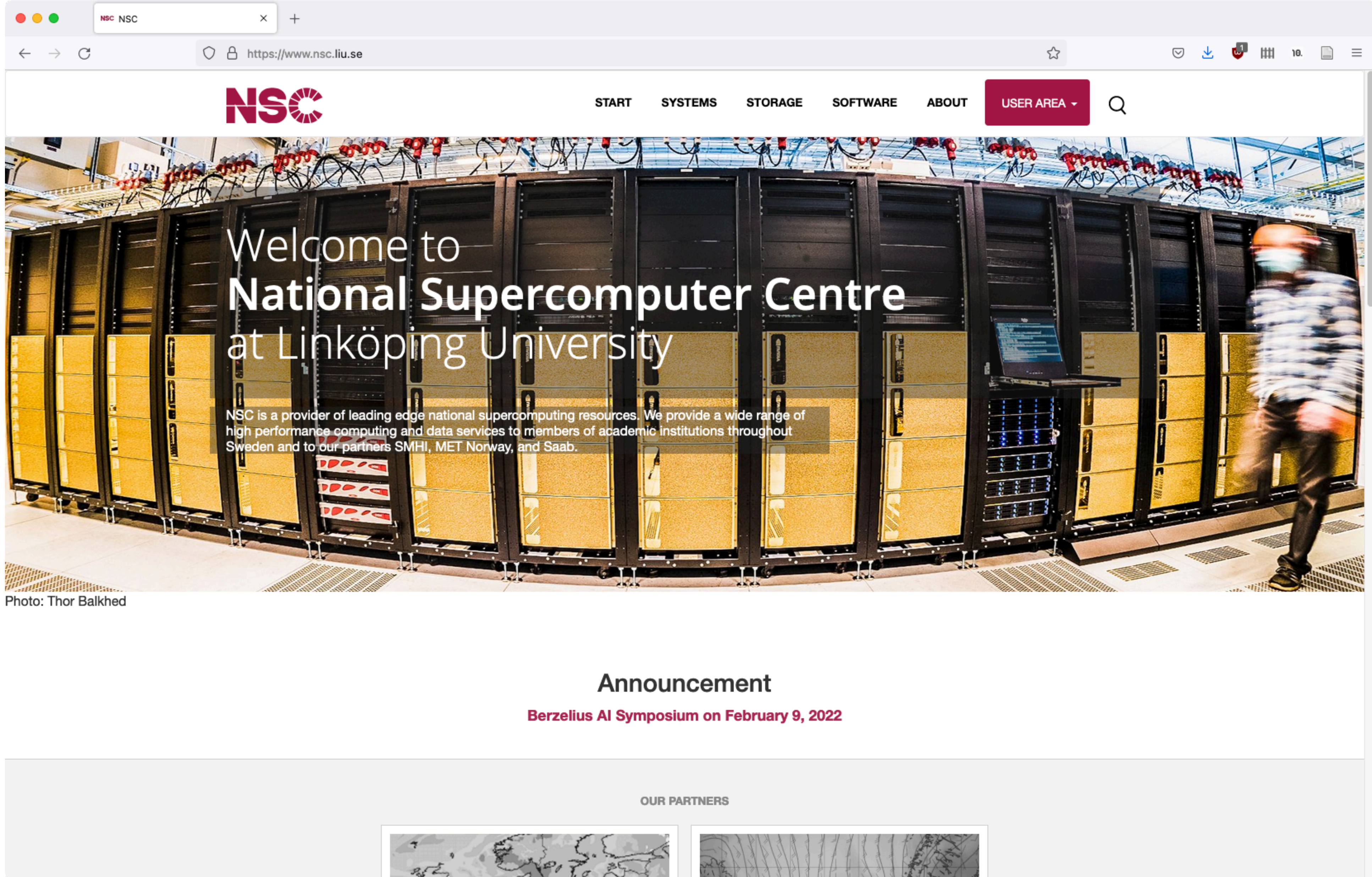


Photo: Thor Balkhed

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



Welcome to National Supercomputer Centre at Linköping University

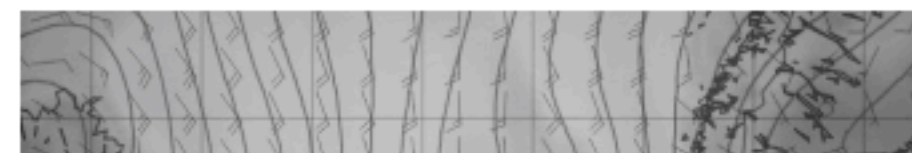
NSC is a provider of leading edge national supercomputing resources. We provide a wide range of high performance computing and data services to members of academic institutions throughout Sweden and to our partners SMHI, MET Norway, and Saab.

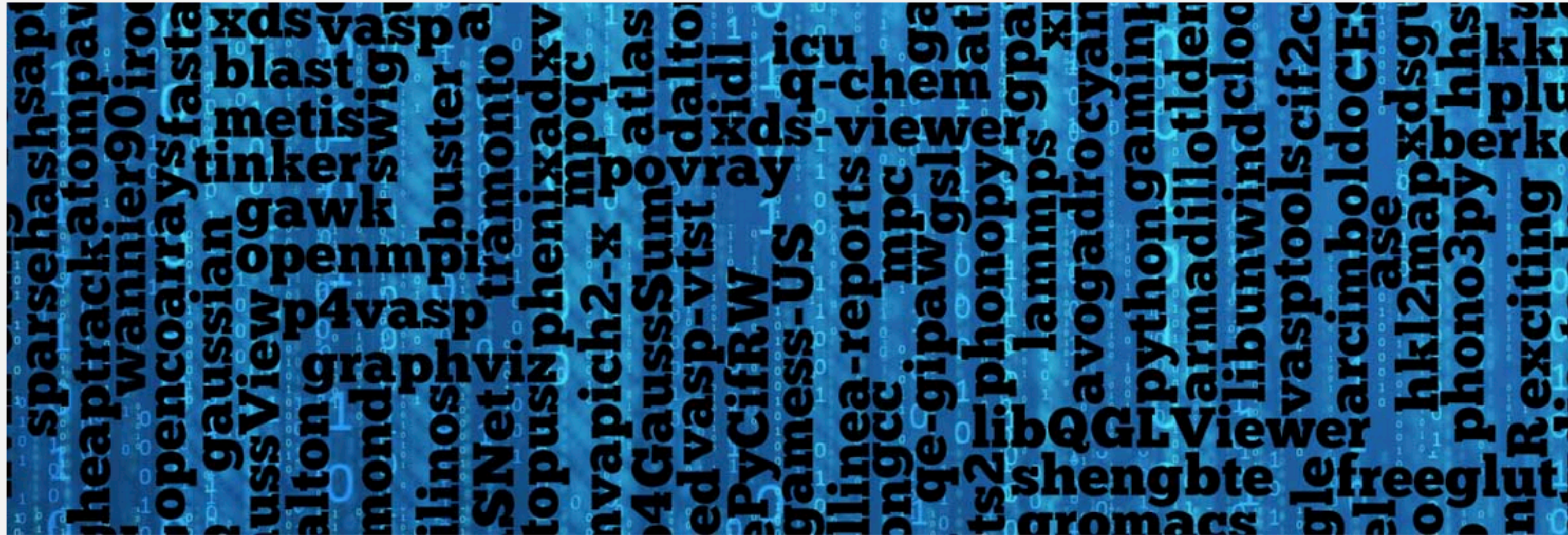
Photo: Thor Balkhed

Announcement

Berzelius AI Symposium on February 9, 2022

OUR PARTNERS





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.

Compilers

Available compilers and recommendations.

NSC build environment

Our recommended way to compile and run your own programs.

Modules

Module system integration at NSC



NSC / Software / Installed 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](#).
- For Bi and Nebula, please look at the list above (software present there that is not already on Bi/Nebula can be requested).

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 and links to further information.



Tetralith & Sigma Software

NSC / Software / Installed software / 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: 2020-03-16

Electronic structure

- [Abinit](#)
- [ASE](#)
- [CASTEP](#)
- [Elk](#)
- [EPW](#)
- [exciting](#)
- [GPAW](#)
- [phonopy](#)
- [phono3py](#)
- [p4vasp](#)
- [Quantum Espresso](#)
- [USPEX](#) (licensed)
- [vasptools](#)
- [VASP](#) (licensed)
- [WIEN2k](#) (licensed)
- [Yambo](#)



- ABAQUS ABINIT AMBER ANSYS ANSYS-EM ASE ATAT Alinea Performance Reports Alinea-DDT
- Alinea/ARM-MAP CASTEP CDO CESM COMSOL CP2K CPMD DL_POLY Dalton/LSDalton EC-Earth
- EPW Eik FERRET GPAW GROMACS Grace Gurobi Optimizer HDF5 Julia LAMMPS MATLAB
- MOLDEN Mathematica NAMD NCO NCVIEW NorESM Open Babel OpenFOAM ParaView Pymatgen
- Quantum ESPRESSO STAR-CCM+ Siesta USPEX VMD Visit WEST WIEN2K Yambo ecCodes exciting
- grib_api netCDF p4vasp parallel phono3py phonopy vaspools Schrödinger suite VASP Clang**
- Gaussian and GaussView

[NSC](#) / [Software](#) / [Installed software](#) / [Tetralith & Sigma Software](#) / 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
```



KTH / PDC / Software

Software

General information about VASP

Licenses

Disclaimer

Installed software

General information about VASP

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.

Disclaimer

PDC takes no responsibility for the correctness of results produced with the binaries. Always evaluate the binaries against known results for the systems and properties you are investigating before using the binaries for production jobs.

Installed software

| Cluster | How to use | How to build |
|---------|--|--|
| Tegner | 5.3.5 5.4.1 | 5.3.5 5.4.1 |



Software

[How to use VASP](#)[General observations](#)[How to choose the number of cores](#)[Parallelization settings](#)[Vasp Filenames](#)[Potential files and vdW kernel](#)[Running Vasp](#)[Disclaimer](#)

How to use VASP

| Software | Version | Cluster |
|----------|---------------|---------|
| VASP | 6.2.1-vanilla | Dardel |

This is a *vanilla* version of VASP 6.2.1, i.e. no extensions have been added to the VASP source code.

For a list of new features in VASP6, see the [VASP wiki](#).

General observations

- VASP is not helped by hyperthreading.
- Running on fewer than 128 tasks 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 out-of-memory (OOM) error. Further information from the VASP wiki can be found [here](#) and [here](#). You can check the example job script for using 64 MPI tasks x 2 OpenMP threads per node uner [Running Vasp](#).

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)
- < 0.5 atom per core = Don't do it

Evaluation of above:

Example of day-to-day tools

- less / gedit / vi / nano reading/editing files
- grace / gnuplot plotting tools
- Bash simple scripts
- cif2cell convert from .cif, create structures
- p4vasp analysis of VASP output
- xcrysden / vesta view structure
- ASE different tools (Python), create structures

Also of interest:

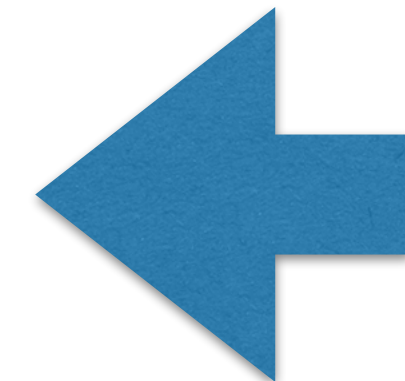
- Python / R analysis etc. (check out e.g. using jupyter)
- Matlab / Octave analysis etc.
- Schrödinger create/view structure @Kebnekaise, Tetralith

[Quick guide to Tetralith for the workshop](#)
(Presentations from old 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 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 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!**
- NSIM - for RMM-DIIS algorithm (ALGO)
- NCORE or NPAR - bands treated in parallel
- KPAR - k-point parallel

parallel
calcs.

- We will get back to the settings in part 3!

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 & moved
no update MD =2 ionic relaxation

POSCAR

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

(hopefully) useful → comment →

lattice vectors →

number of atoms
per type →

position for first atom →

```
Ni fcc
3.53
0.5 0.5 0.0
0.0 0.5 0.5
0.5 0.0 0.5
Ni
1
Cartesian
0 0 0
```

← lattice constant (Å)

← element symbols

← Cartesian or Direct coordinates

negative value:
cell volume

optional, useful for
clarity & plotting

First letter is sufficient, i.e.
“C” for “Cartesian”

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

1.0000000000000000 0.0000000000000000 0.0000000000000000

0.5000000000000000 0.866025403784439 0.0000000000000000

0.0000000000000000 0.0000000000000000 4.352531500114909

H Si Ag

14 14 108

← Note order of atoms

Selective dynamics

← Relax for different directions

Direct

first H atom →

0.7583380000000000 0.0528816000000000 0.6059450000000000

0.5052440000000000 0.1182310000000000 0.6059450000000000

0.0507845000000000 0.1960170000000000 0.6059450000000000

0.8003590000000000 0.2519030000000000 0.6059450000000000

0.3333333333333333 0.3389020000000000 0.6059450000000000

0.1141910000000000 0.3862440000000000 0.6059450000000000

0.6184740000000000 0.4980580000000000 0.6059450000000000

0.3812620000000000 0.5113130000000000 0.6059450000000000

0.8830710000000000 0.6246840000000000 0.6059450000000000

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

Rest of H,
Si & Ag atoms
following



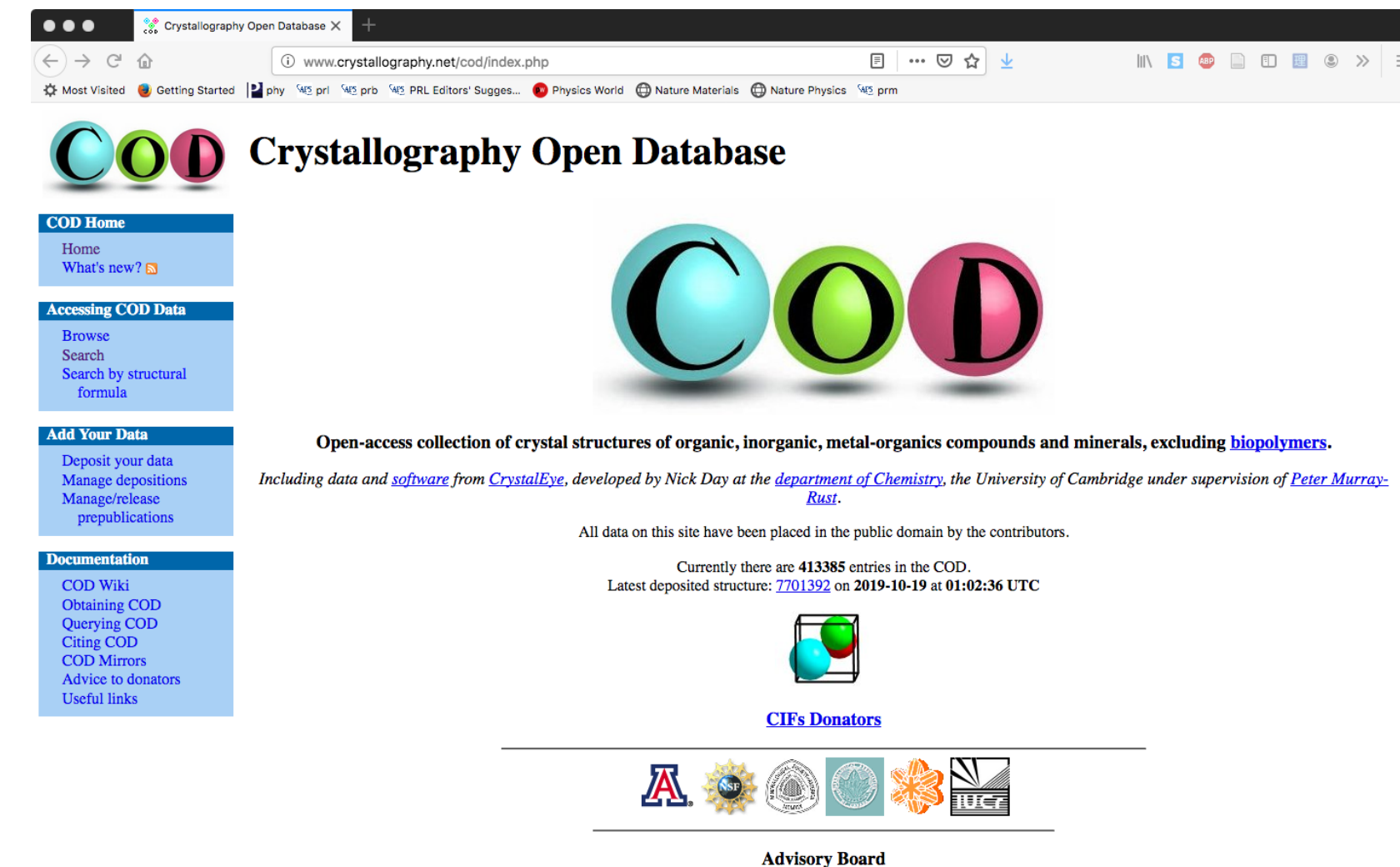
T T T
T T T
T T T
T T T
T T T
T T T
T T T
T T T

POSCAR

Some useful resources:

[Crystallography Open Database](http://www.crystallography.net/cod/index.php)

Database with published structures from experiment .cif



Crystallography Open Database

Open-access collection of crystal structures of organic, inorganic, metal-organics compounds and minerals, excluding biopolymers. Including data and software from CrystalEye, developed by Nick Day at the department of Chemistry, the University of Cambridge under supervision of Peter Murray-Rust.

All data on this site have been placed in the public domain by the contributors.

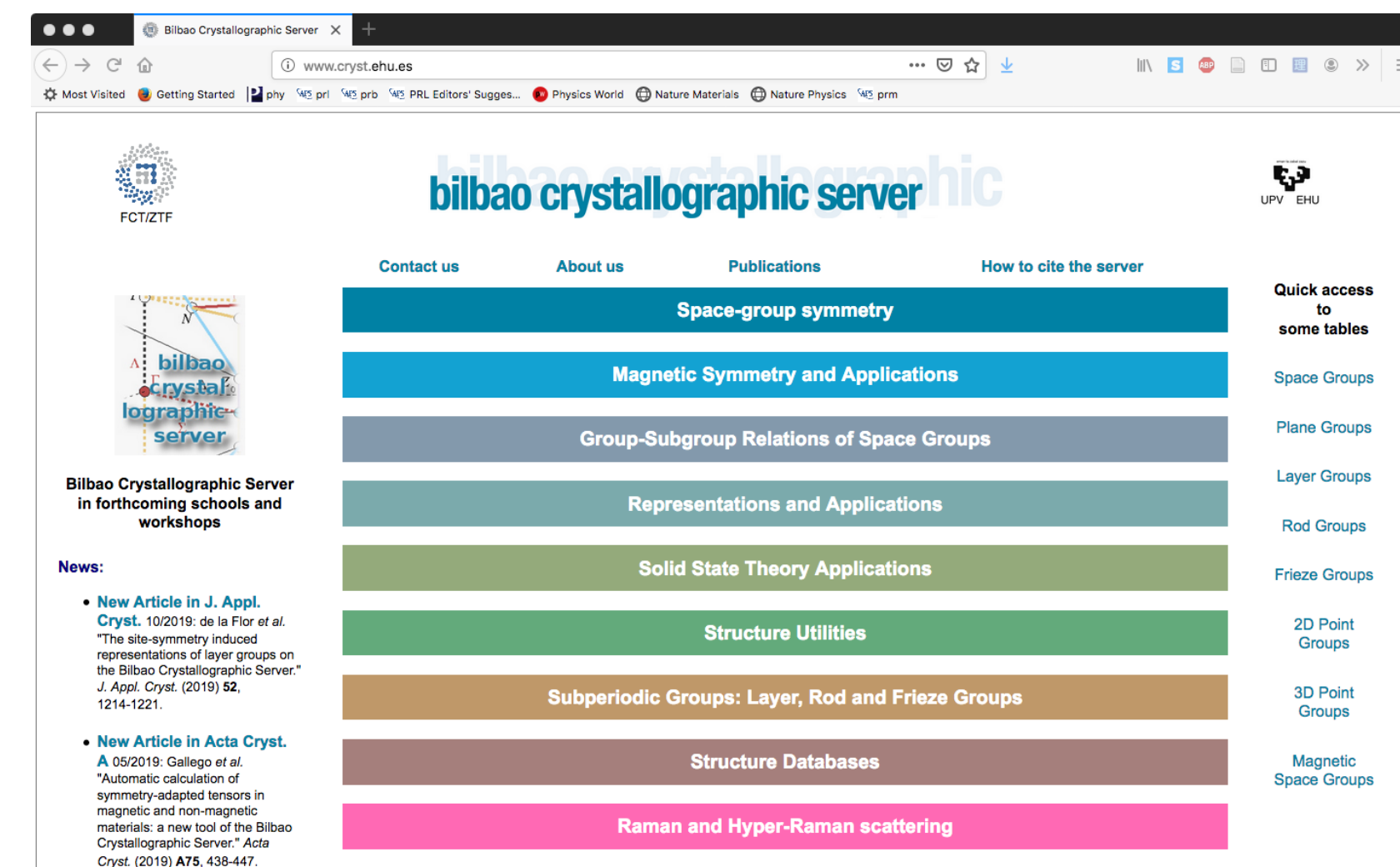
Currently there are 413385 entries in the COD.
Latest deposited structure: 7201392 on 2019-10-19 at 01:02:36 UTC

CIFs Donators

Advisory Board

[Bilbao Crystallographic Server](http://www.cryst.ehu.es)

Many Crystallographic tools, e.g. check BZ of fcc cell



bilbao crystallographic server

Contact us About us Publications How to cite the server

Space-group symmetry

Magnetic Symmetry and Applications

Group-Subgroup Relations of Space Groups

Representations and Applications

Solid State Theory Applications

Structure Utilities

Subperiodic Groups: Layer, Rod and Frieze Groups

Structure Databases

Raman and Hyper-Raman scattering

Quick access to some tables

Space Groups

Plane Groups

Layer Groups

Rod Groups

Frieze Groups

2D Point Groups

3D Point Groups

Magnetic Space Groups

News:

- New Article in J. Appl. Cryst. 10/2019: de la Flor et al. "The site-symmetry induced representations of layer groups on the Bilbao Crystallographic Server." J. Appl. Cryst. (2019) 52, 1214-1221.
- New Article in Acta Cryst. A 05/2019: Gallego et al. "Automatic calculation of symmetry-adapted tensors in magnetic and non-magnetic materials: a new tool of the Bilbao Crystallographic Server." Acta Cryst. (2019) A75, 438-447.

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

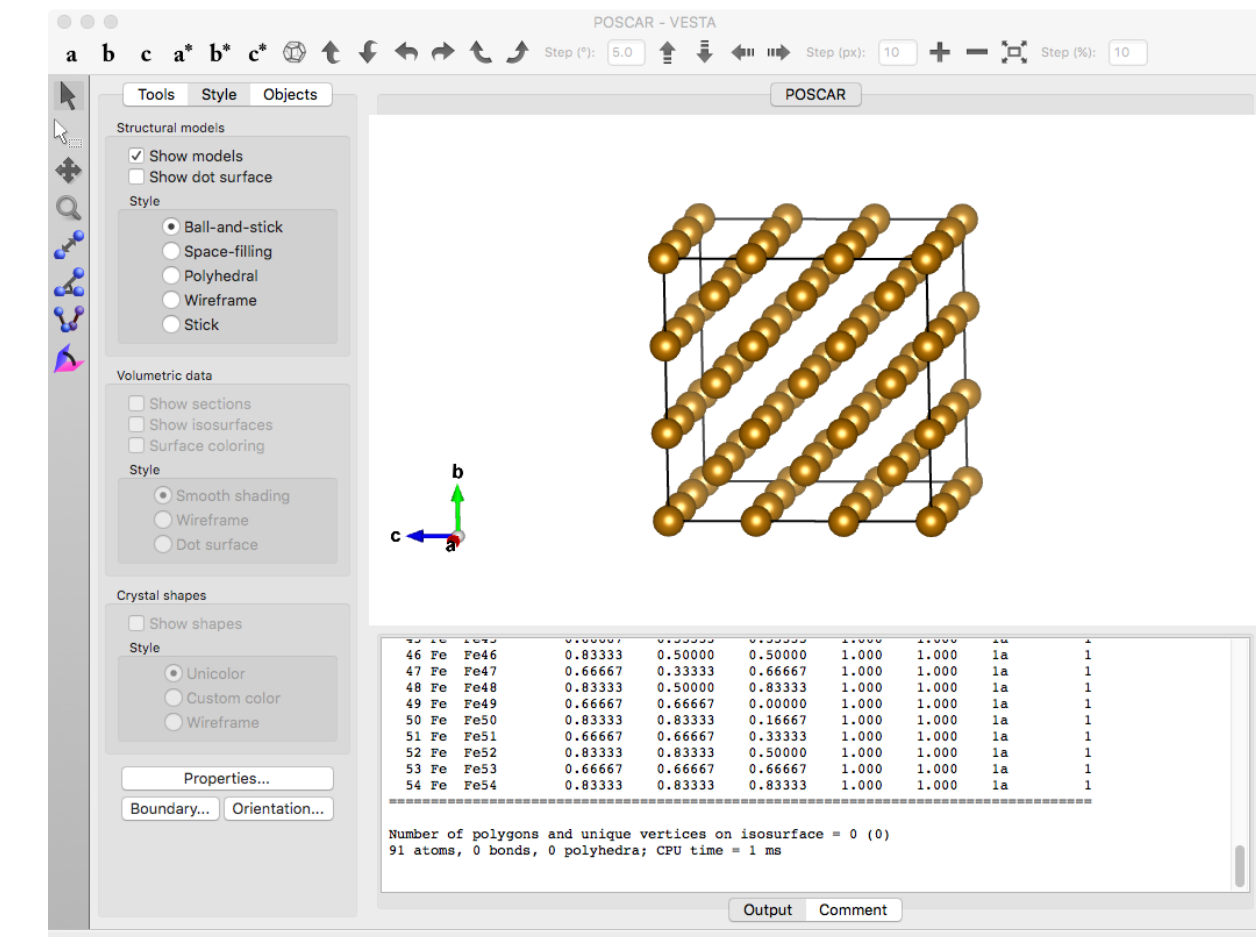
Commercial software:

[NanoLab](#)

[MedeA](#)

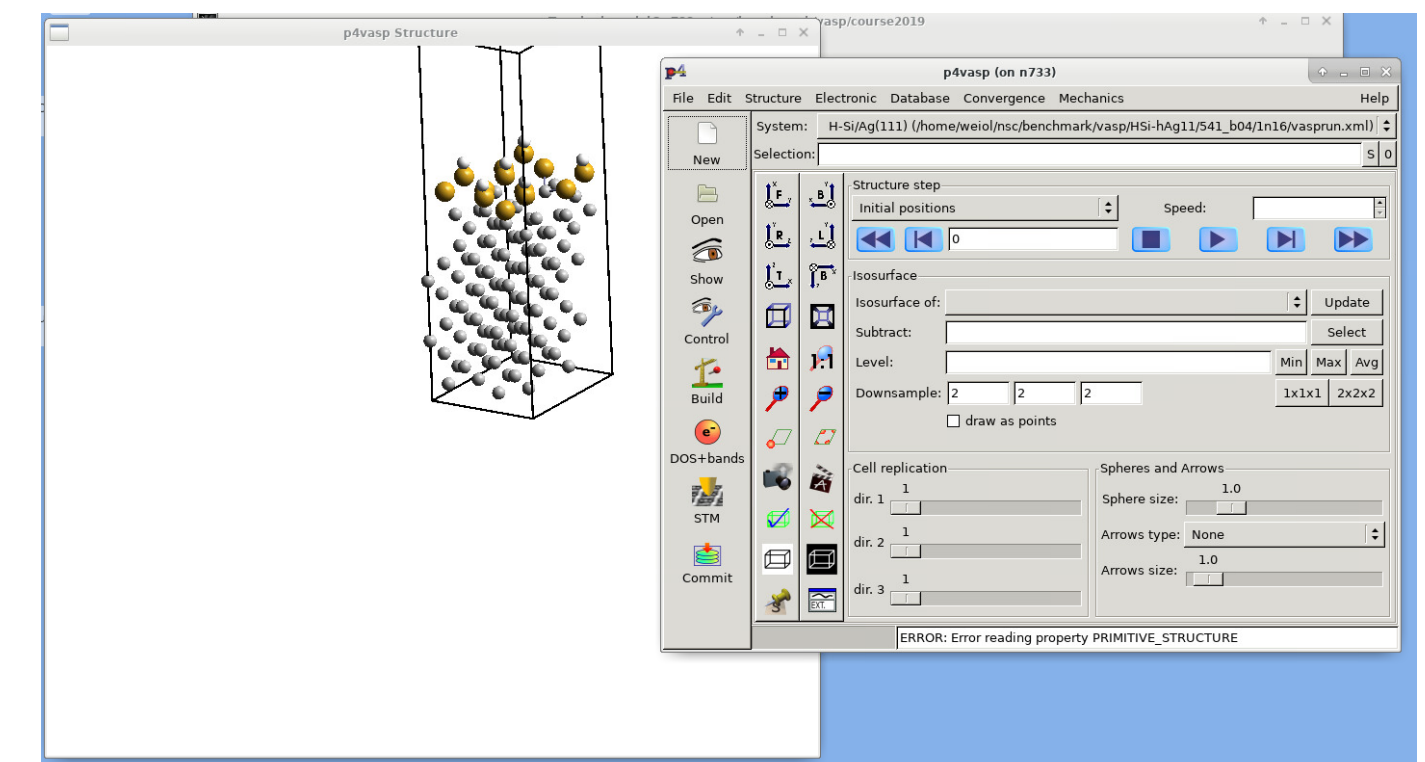
[MaterialsStudio](#)

[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 → 11.000000000000000
parameters from PSCTR are:
XR-type → VRHFIN =Cu: d10 p1 ← atomic configuration
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

Note several choices, e.g.:

Ga, **Ga_d**, Ga_d_GW,
Ga_GW, Ga_h, Ga_sv_GW

- for GW: `_GW`

- States in valence: `_sv`, `_pv`, `_d`

Useful commands:

```
$ grep PAW POTCAR
```

```
$ grep ENMAX POTCAR
```

- “soft” (no short bonds): `_s`

- Where?

[@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

0 = automatic generation of mesh →

```
k-points ← comment
0
Monkhorst Pack ← Monkhorst-Pack method (M)
11 11 11 ← odd kmesh - includes  $\Gamma$ -point
0 0 0 ← optional shift of k-mesh
```

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

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

```
Automatic mesh
0
Gamma ← Gamma method (G)
2 2 1
0. 0. 0.
```

- 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 ← comment
k-points per line-segment → 10
line ← k-points per line-segment
Reciprocal / Cartesian → Reciprocal
0.50000 0.50000 0.50000 1 ← symmetry point + weight
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
```

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
- `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

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)

#SBATCH -n 64

note "mpprun"

```
#!/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
```

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)

Note:
Beskow is now retired!

```
#!/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>

Job script - Dardel (PDC)

- I'll test running
VASP @Dardel
in near future

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

module load PDC/21.11
module load vasp/6.2.1-vanilla

export OMP_NUM_THREADS=1

srun vasp
```

Example: running on 2 nodes (128x2 cores) @Dardel

https://www.pdc.kth.se/software/software/VASP/cpe21.11/6.2.1-vanilla/index_using.html

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
output can also be switched off
- [WAVECAR](#) - plane wave coefficients (for restart)
- ...

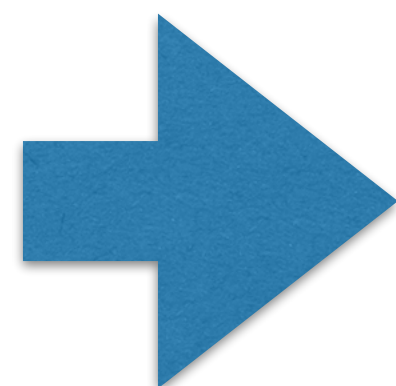
OSZICAR

| 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
```

- Check for warnings!

| | 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
```

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   AAAAA   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.)

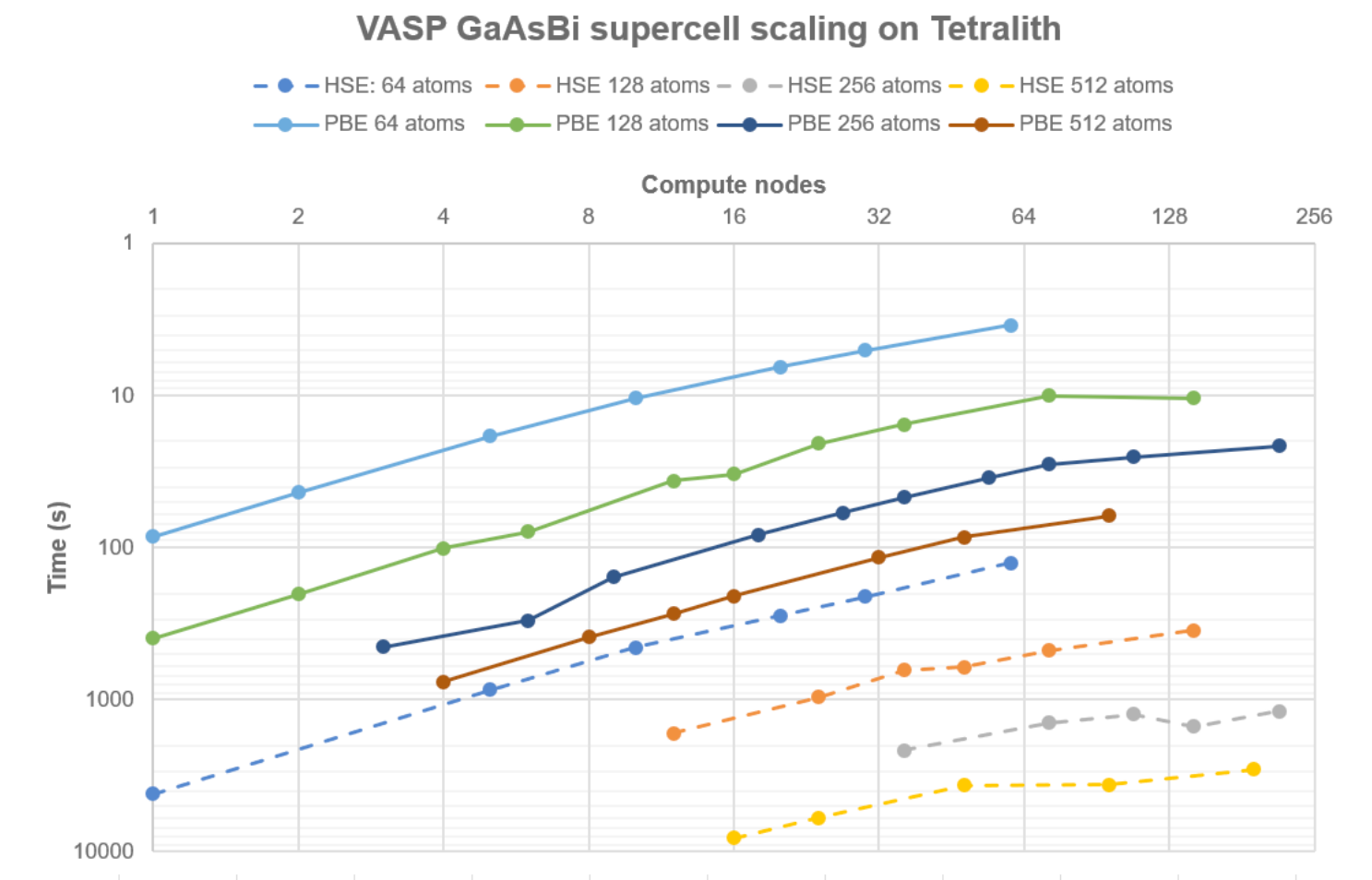
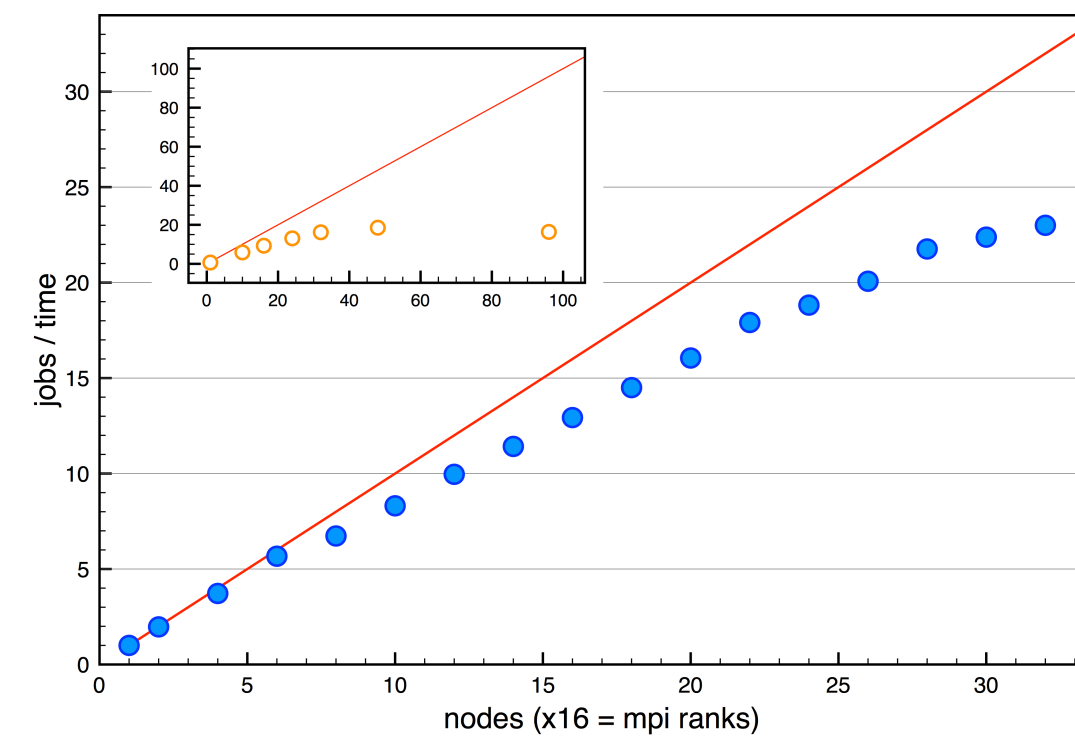
VASP6

- [Link to features in VASP6.X](#)
- 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

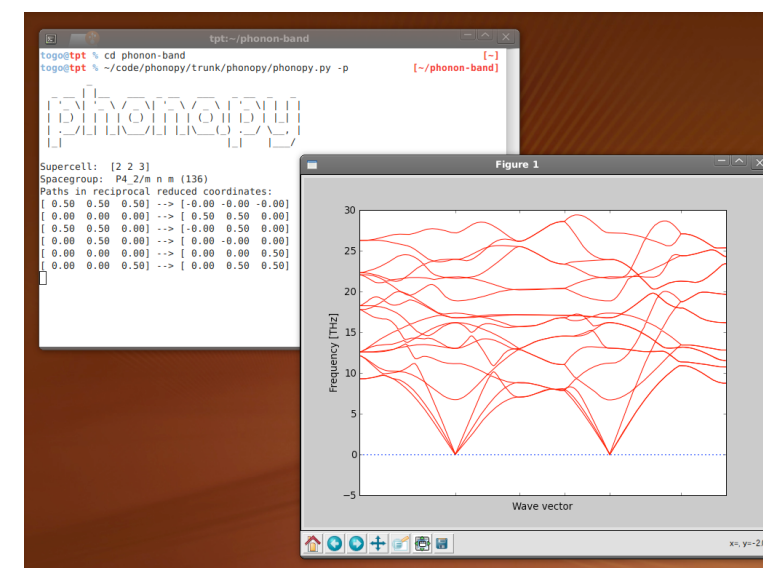
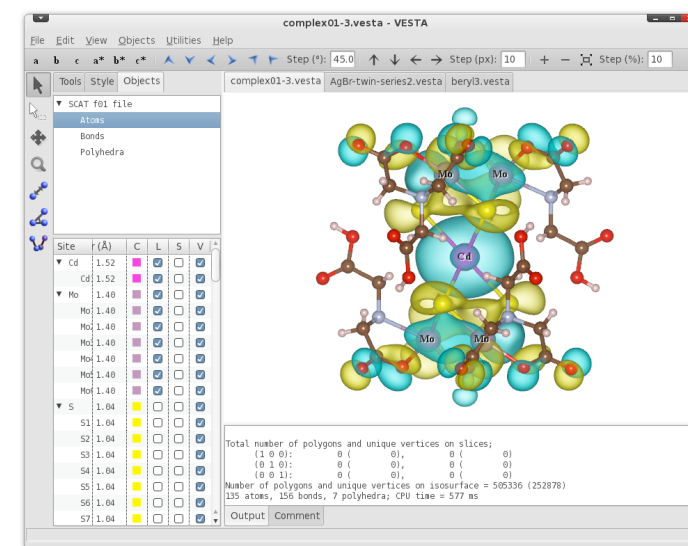
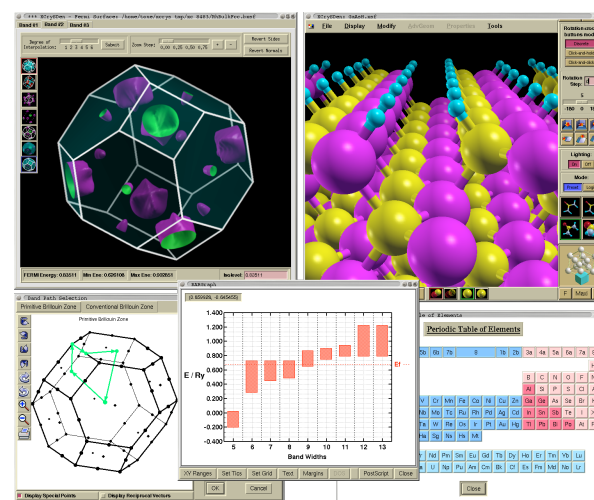
Tomorrow:

- Running & Performance

- Parallelization
- Efficient settings
- **Problems**



- Utilities & Summary



VTST•Tools

USPEX Computational Materials Discovery