

# 2. VASP - Basics

Weine Olovsson

National Supercomputer Centre (NSC), Linköping University

NAISS-ENCCS training, online 17-18<sup>th</sup> Apr 2024

# VASP - Best Practices Workshop



**NAISS**



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

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

<https://encacs.se/>



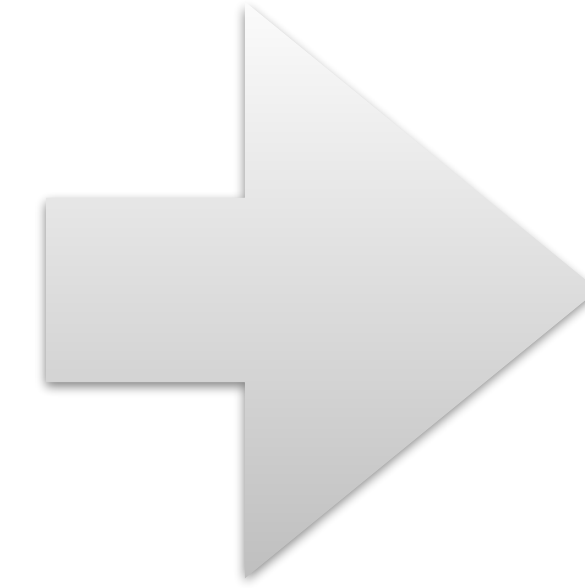
# Introduction

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

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

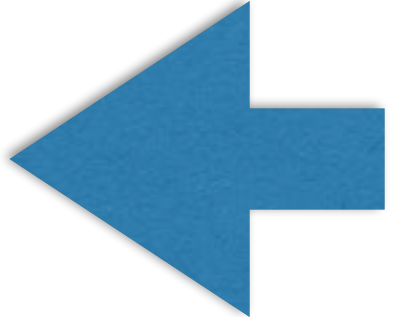
# Short background

- 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

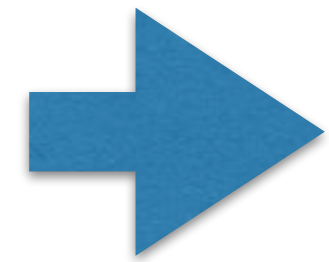


<https://vasp.at/>

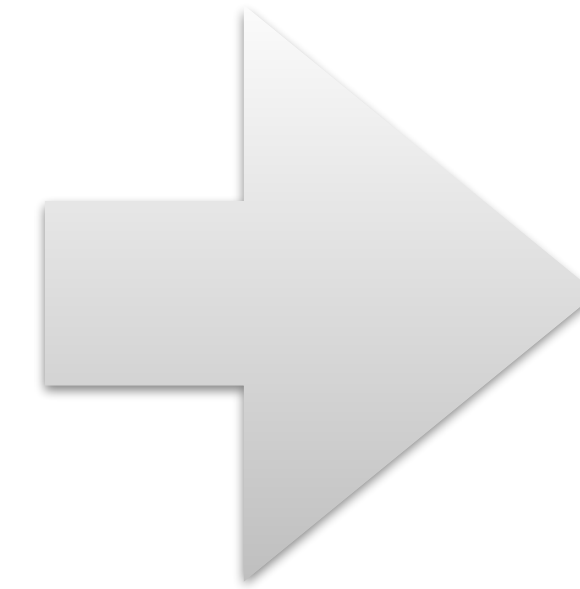
# Starting advice

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

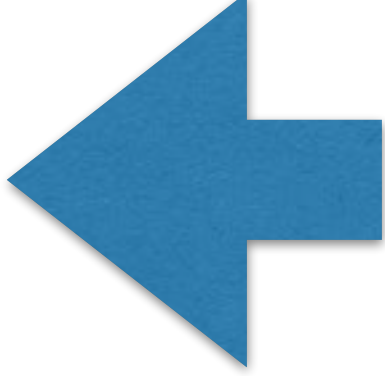
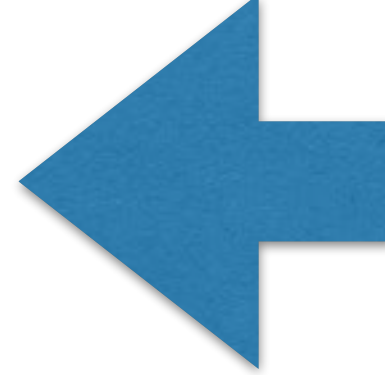


Find all the links:  
<https://vasp.at/>

- Also other resources, materials and tools for VASP (see presentation 4.)
- Peter Larsson's old blog at NSC: <https://www.nsc.liu.se/~pla/>
- NSC VASP installations: <https://www.nsc.liu.se/software/installed/tetralith/vasp/>

**Questions / trouble @NSC clusters? [support@nsc.liu.se](mailto:support@nsc.liu.se)**

# VASP & HPC centers

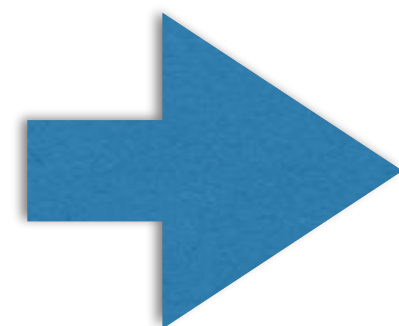
- **Tetralith (NAISS)** / Sigma, NSC, LiU  **this course**  
**Installed!**
- Dardel (NAISS), PDC, KTH  
**Installed!**
- Other systems/centers (e.g. Kebnekaise, HPC2N)  
**Installed!**
- LEONARDO (EuroHPC) - workshop  **this course**
- LUMI (EuroHPC & NAISS)

- **Check for modules:** \$ module avail vasp  
\$ module spider vasp

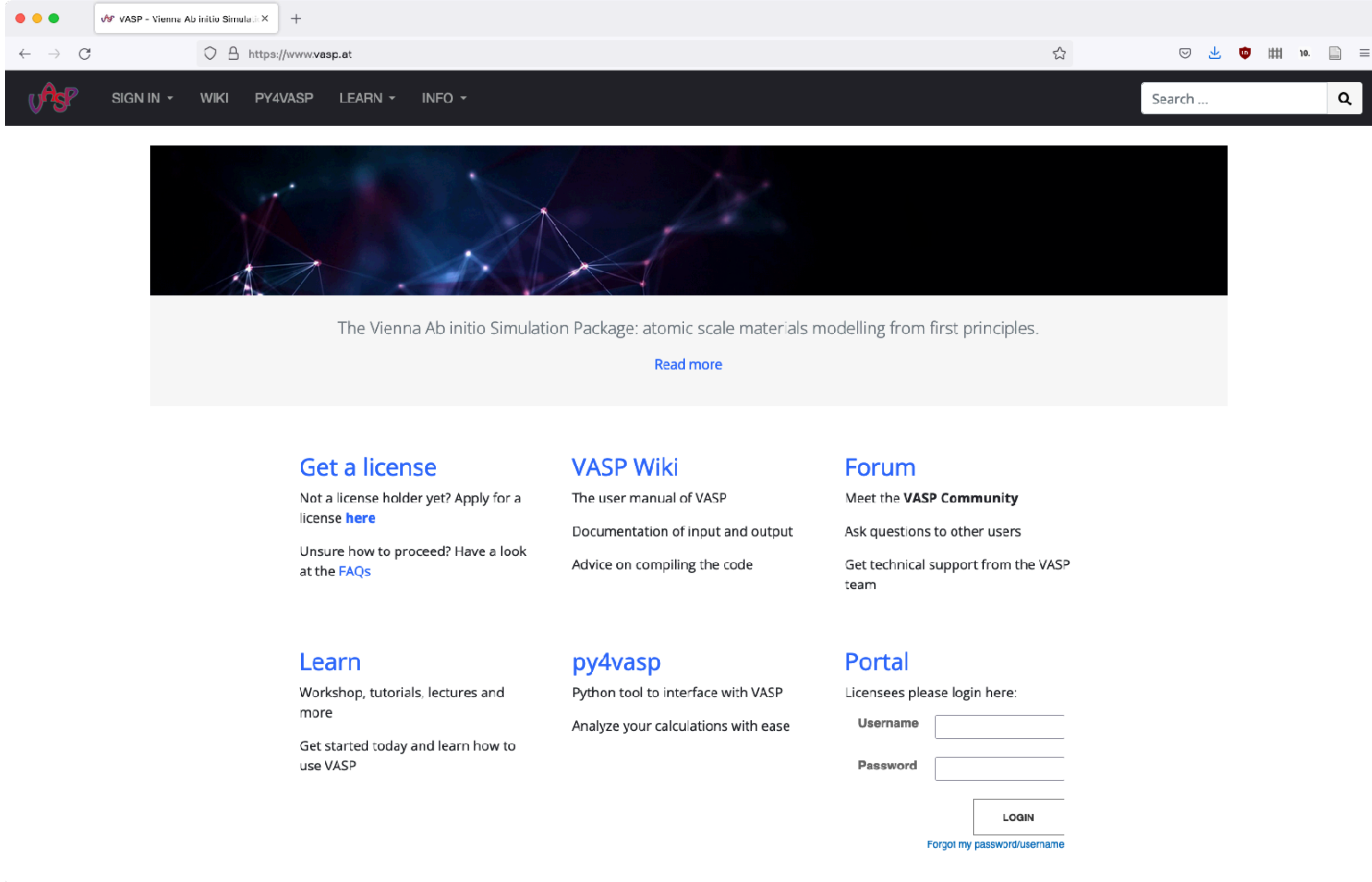


# VASP versions & utilities

- **Latest:** 6.4.3 (from March -24)  
- 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



...more about utilities tomorrow



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

...the official VASP webpage

- 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. <sup>1</sup> 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](#)

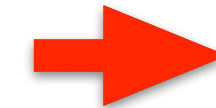


### VASP Wiki

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](#)

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

## The VASP Manual

### Getting started

<a href="#">How to Install VASP</a>	First install VASP.
<a href="#">Learn</a>	Tutorials using <a href="#">notebooks</a> , <a href="#">video lectures</a> , etc.
<a href="#">More</a>	How to's, <a href="#">examples</a> and <a href="#">tutorials</a> on the Wiki, some <a href="#">lectures</a> and <a href="#">schedules</a> from previous workshops, etc.
<a href="#">VASP6</a>	Features that will only be available in VASP.6.X.

### Featured topics

Category	subtopics (amongst others)
<a href="#">Theoretical background</a>	Density-functional theory, projector-augmented-wave method, molecular dynamics, GW approximation, etc.
<a href="#">Calculation setup</a>	Installation, <a href="#">input files</a> , <a href="#">output files</a> , INCAR tags, How to's, etc.
<a href="#">Electronic minimization</a>	Davidson, RMM-DIIS, Conjugate-gradient, preconditioning, <a href="#">density mixing</a> , etc.
<a href="#">Electronic ground-state properties</a>	Band structure, <a href="#">density of states</a> , etc.
<a href="#">Spin degree of freedom</a>	Spin-orbit coupling, noncollinear magnetism, spin spirals, constrained magnetism, etc.
<a href="#">Exchange-correlation functionals</a>	LDA, GGA, meta-GGA, DFT+U, <a href="#">hybrid functionals</a> , <a href="#">van der Waals functionals</a> .
<a href="#">Symmetry and structure</a>	Crystal symmetry, reciprocal space, surfaces, pair-correlation function for liquids, etc.
<a href="#">Ionic mimimization</a>	Structure optimization, ionic-mimimization methods, <a href="#">forces</a> , etc.

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](https://www.vasp.at/wiki/index.php/The_VASP_Manual)

The VASP Manual - Vaspwiki

https://www.vasp.at/wiki/index.php/The\_VASP\_Manual

<a href="#">Symmetry and structure</a>	Crystal symmetry, reciprocal space, surfaces, pair-correlation function for liquids, etc.
<a href="#">Ionic minimization</a>	Structure optimization, ionic-minimization methods, <a href="#">forces</a> , etc.
<a href="#">Molecular dynamics</a>	Barostats, <a href="#">thermostats</a> , <a href="#">ensembles</a> , etc.
<a href="#">Ensemble properties</a>	Monitoring geometric parameters, pair-correlation function, etc.
<a href="#">Advanced molecular-dynamics sampling</a>	<a href="#">Interface pinning</a> , constrained molecular dynamics, metadynamics, thermodynamic integration, etc.
<a href="#">Machine-learned force fields</a>	Training and application of force fields.
<a href="#">Phonons</a>	Lattice vibrations, <a href="#">electron-phonon interactions</a> .
<a href="#">Response theory</a>	Static and frequency-dependent <a href="#">dielectric properties</a> , Berry phases, spectroscopy (UV, VIS, X-ray), <a href="#">phonons</a> , etc.
<a href="#">Many-body perturbation theory</a>	ACFDT, BSE, GW, MP2, CRPA.
<a href="#">Localized basis and projection</a>	Obtaining Wannier functions, SCDM, etc.
<a href="#">Performance</a>	<a href="#">Parallelization</a> , <a href="#">memory management</a> , profiling, etc.
<a href="#">Outlook</a>	Features that will only be available in VASP.6.X.

## Support

If you have questions or run into trouble, please have a look at the [known issues](#) and/or post a question at the [VASP forum](#).



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

[Back to the top](#)

This page was last edited on 12 August 2022, at 09:58.

Content is available under [GNU Free Documentation License 1.2](#) unless otherwise noted.

[Privacy policy](#) [About Vaspwiki](#) [Disclaimers](#)

 GNU FDL  
FREE DOC  
LICENSE  Powered By  
MediaWiki

[https://www.vasp.at/wiki/index.php/The\\_VASP\\_Manual](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.

- A**
  - Adsorption of H2O on TiO2
  - Alpha-AlF3
  - Alpha-SiO2
  - At and mol further
- B**
  - 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
- C**
  - Calculate U for LSDA+U
  - Cd Si
  - Cd Si relaxation
  - Cd Si volume relaxation
  - CO
  - CO on Ni 111 surface
  - CO partial DOS
- E**
  - Estimation of J magnetic coupling
- F**
  - Fcc Ni
  - Fcc Ni (revisited)
  - Fcc Ni DOS
  - Fcc Ni DOS with hybrid functional
  - Fcc Si
  - Fcc Si bandstructure
  - Fcc Si DOS
- G**
  - Graphite interlayer distance
  - Graphite MBD binding energy
  - Graphite TS binding energy
- H**
  - H2O
  - H2O molecular dynamics
  - H2O vibration
- I**
  - Improving the dielectric function
  - Including the Spin-Orbit Coupling
- N**
  - 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
- O**
  - O atom
  - O atom spinpolarized
  - O atom spinpolarized low symmetry
  - O dimer
- P**
  - Partial DOS of CO on Ni 111 surface
  - Plotting the BSE fatband structure of Si
- S**

[https://www.vasp.at/wiki/index.php/The\\_VASP\\_Manual](https://www.vasp.at/wiki/index.php/The_VASP_Manual)

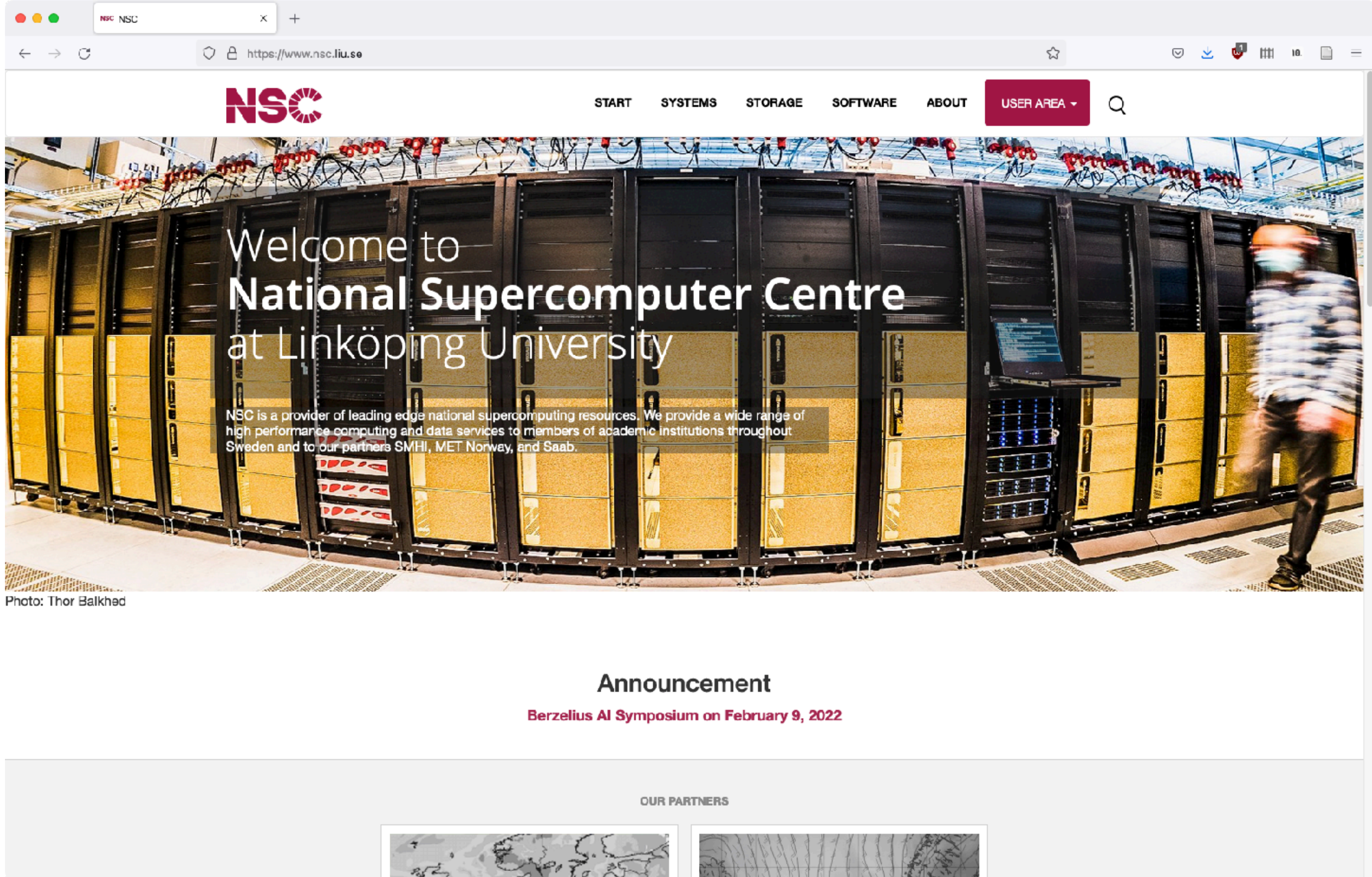


Photo: Thor Balkhed

<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

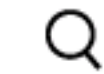
Available compilers and recommendations.

### NSC build environment

Our recommended way to compile and run your own programs.

### Modules

Module system integration at NSC



## 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.

## Software on Tetralith and Sigma

The scientific applications listed in the table below have been installed centrally under </software/sse/>. Each software installation is categorized into one of three [software support tiers](#), depending on the level of help we can provide for that particular software.

Some useful software and tools, such as a few editors are installed as part of the operating system and are not listed here.

This list was last updated on: 2023-03-20

- All
- Chem
- Phys
- Bio
- CAE
- Geo
- Math
- Devel
- Data
- Tools
- Vis
- Misc

Search:

Software <span style="float: right;">↑↓</span>	Description <span style="float: right;">↑↓</span>	Support <span style="float: right;">↑↓</span>
<a href="#">ABAQUS</a>	The Abaqus FEA software suite offers various tools for stress analysis, heat transfer, fluid mechanics etc.	tier3
<a href="#">ABINIT</a>	ABINIT calculates the total energy and properties of materials and molecules using, primarily, DFT.	tier2
<a href="#">allinea-DDT</a>	ARM/Allinea DDT is a debugging tool for scalar, multi-threaded and large-scale parallel applications.	tier2
<a href="#">allinea-forge</a>	ARM/Allinea DDT is a debugging tool for scalar, multi-threaded and large-scale parallel applications.	tier2
<a href="#">allinea-MAP</a>	ARM/Allinea MAP is a profiler for scalar, multi-threaded and large-scale parallel applications.	tier2
<a href="#">AlphaFold</a>	Implementation of the inference pipeline of AlphaFold v2.0.	tier3
<a href="#">Amber</a>	Amber is a suite of biomolecular simulation programs.	tier3
<a href="#">Anaconda</a>	The Anaconda Python distribution platform for Python and R scientific computing.	tier2
<a href="#">ANSYS</a>	Software suite for computational fluid dynamics, stress analysis and many more CAE tasks.	tier2
<a href="#">ANSYS-EM</a>	Software suite for Electromagnetics simulations.	tier3

## Software on Tetralith and Sigma

The scientific applications listed in the table below have been installed centrally under </software/sse/>. Each software installation is categorized into one of three [software support tiers](#), depending on the level of help we can provide for that particular software.

Some useful software and tools, such as a few editors are installed as part of the operating system and are not listed here.

This list was last updated on: 2023-03-20

[All](#) [Chem](#) [Phys](#) [Bio](#) [CAE](#) [Geo](#) [Math](#) [Devel](#) [Data](#) [Tools](#) [Vis](#) [Misc](#)

Search:

Software	Description	Support
<a href="#">p4vasp</a>	p4vasp is a popular tool for VASP visualization.	tier3
<a href="#">VASP</a>	Electronic structure calculations using PAW-method DFT	tier1
<a href="#">VASP-OMC</a>	Modified VASP for occupation matrix control	tier3
<a href="#">VASP-VTST</a>	Utility version of VASP including: VTST 3.2, VASPsol, BEEF	tier3
<a href="#">vasptools</a>	A collection of useful VASP scripts.	tier3

Showing 5 entries (filtered from 163 total entries)



**National Supercomputer Centre**  
Linköping University  
581 83 LINKÖPING  
SWEDEN

E-mail: [support@nsc.liu.se](mailto:support@nsc.liu.se)  
Tel.: 013-281000 (switchboard)  
Fax.: 013-149403  
Further address information

NSC is part of Linköping University and the Swedish National Infrastructure for Computing (SNIC).



Org.nr: 202100-3096  
VAT.nr: SE202100309601

[Top of Page](#)

# VASP

**i** This software is under [support tier 1!](#)

## Software description

VASP - Vienna ab initio simulation package, "a computer program for atomic scale materials modelling, e.g. electronic structure calculations and quantum mechanical molecular dynamics, from first principles."

## License

Note: VASP is a licensed software. Please check further information on the licensing page: [www.nsc.liu.se/software/software-licensing/vasp/](http://www.nsc.liu.se/software/software-licensing/vasp/)

Homepage: [vasp.at/](http://vasp.at/)

## NSC documentation

**i** Please make sure to read our documentation: [VASP](#)

## Available Modules

- [6.3.2.27062022-omp-nsc1-intel-2018a-eb](#)
- [6.3.1.04052022-omp-nsc1-intel-2018a-eb](#)
- [6.3.0.20012022-omp-nsc1-intel-2018a-eb](#)
- [6.2.1.29042021-omp-nsc1-intel-2018a-eb](#)
- [6.2.0.14012021-omp-nsc1-intel-2018a-eb](#)
- [6.1.2.25082020-omp-nsc1-intel-2018a-eb](#)
- [6.1.2.25082020-nsc1-intel-2018a-eb](#)
- [6.1.0.28012020-nsc1-intel-2018a-eb](#)
- [5.4.4.16052018-wannier90-nsc2-intel-2018a-eb](#)
- [5.4.4.16052018-wannier90-nsc1-intel-2018a-eb](#)
- [5.4.4.16052018-vanilla-nsc1-intel-2018a-eb](#)
- [5.4.4.16052018-nsc2-intel-2018a-eb](#)
- [5.4.4.16052018-nsc1-intel-2018b-eb](#)



- ABAQUS ABINIT AMBER ANSYS ANSYS-EM ASE ATAT Allinea Performance Reports Allinea-DDT
- Allinea/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 UppASD 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 / / VASP

## VASP Installations on Tetralith & Sigma

First of all, **VASP** is a 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.

### VASP6

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 5.4.4 license, you are typically covered for updates of VASP 6.X.X for three years, check your license for the exact details.

### Documentation

For using VASP, refer to [the extensive VASP wiki](#) which includes many examples and topics on how to run VASP. It can also be very useful to check the forum and other resources available at [the VASP webpage](#).

The new features of VASP6 are described in [the VASP wiki](#).

From time to time NSC will arrange seminars or brief workshops on VASP with focus typically on how to get started using



- 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 vasptools 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	<a href="#">5.3.5</a> <a href="#">5.4.1</a>	<a href="#">5.3.5</a> <a href="#">5.4.1</a>



## 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:

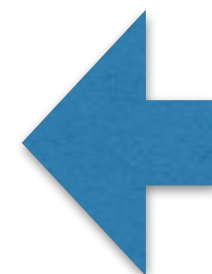
# Example of day-to-day tools

- 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:

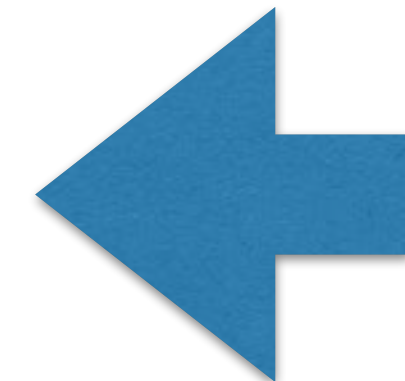
- Python / R      analysis etc. (check out e.g. using jupyter)
- Matlab / Octave      analysis etc.
- Schrödinger      create/view structure      @Tetralith

[Quick guide to Tetralith for the workshop](#)  
(Presentations from NSC Tetralith training)



# Different types of calculations

- Structural relaxation (different ways)
- Regular  $E_{\text{tot}}$  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!**

---

- 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  $E_{\text{tot}}$
- 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
Selective dynamics
Direct
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

first H atom →

← Note order of atoms

← Relax for different directions

Rest of H,  
Si & Ag atoms  
following

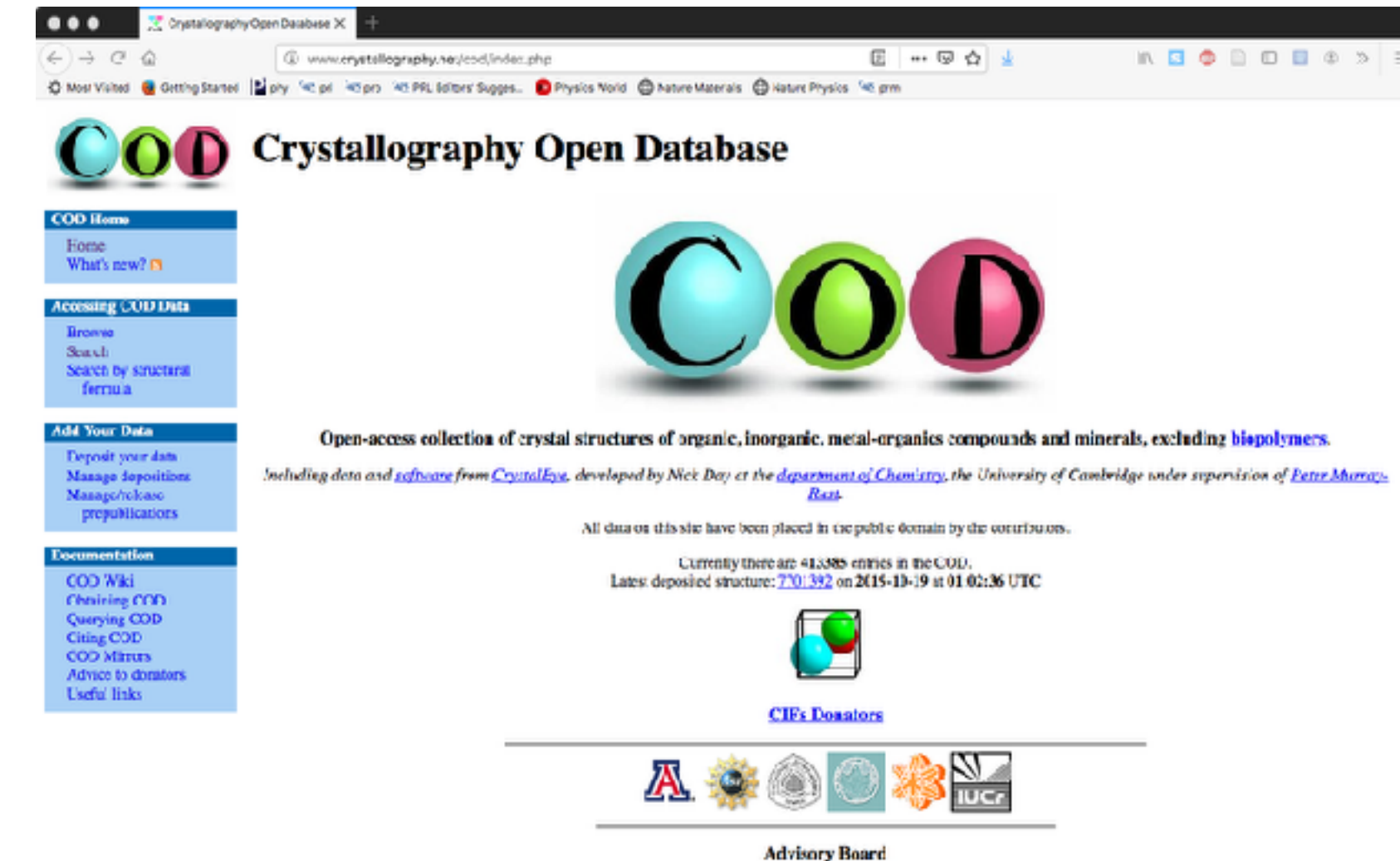


# POSCAR

Some useful resources:

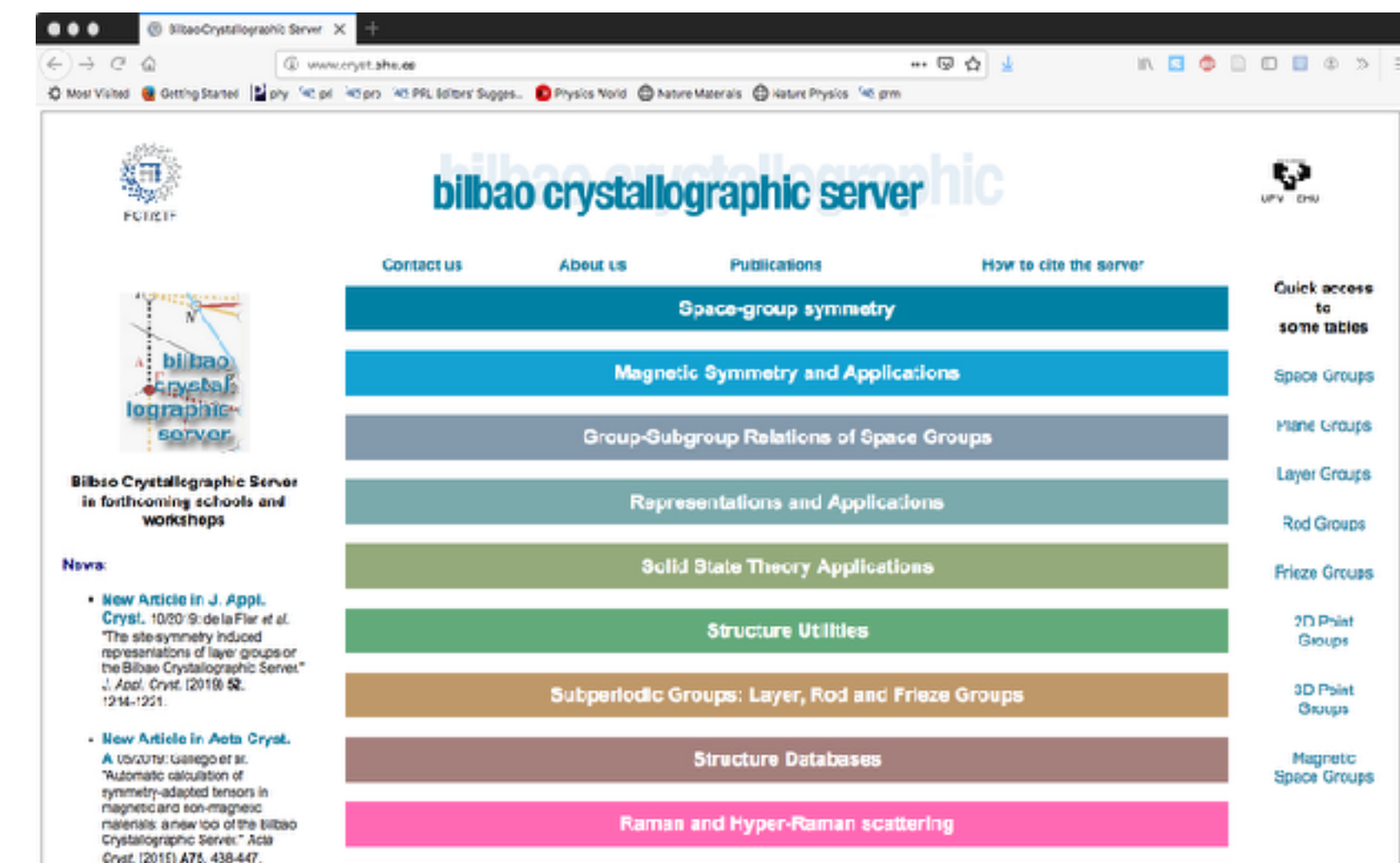
## [Crystallography Open Database](http://www.crystallography.net/cod/index.php)

Database with published structures from experiment .cif



## [Bilbao Crystallographic Server](http://www.crytet.she.ac.uk)

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

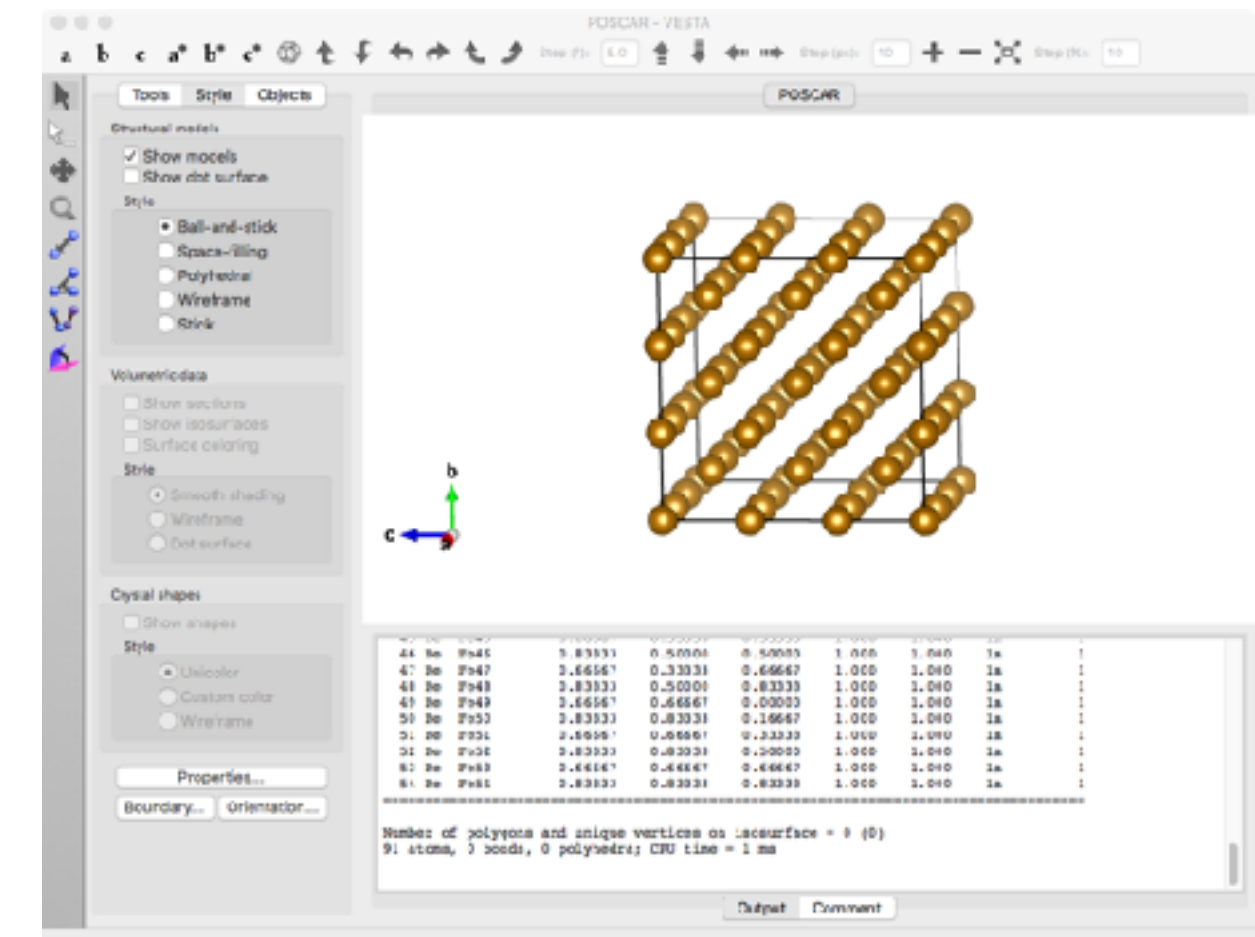
**Commercial software:**

[NanoLab](#)

[MedeA](#)

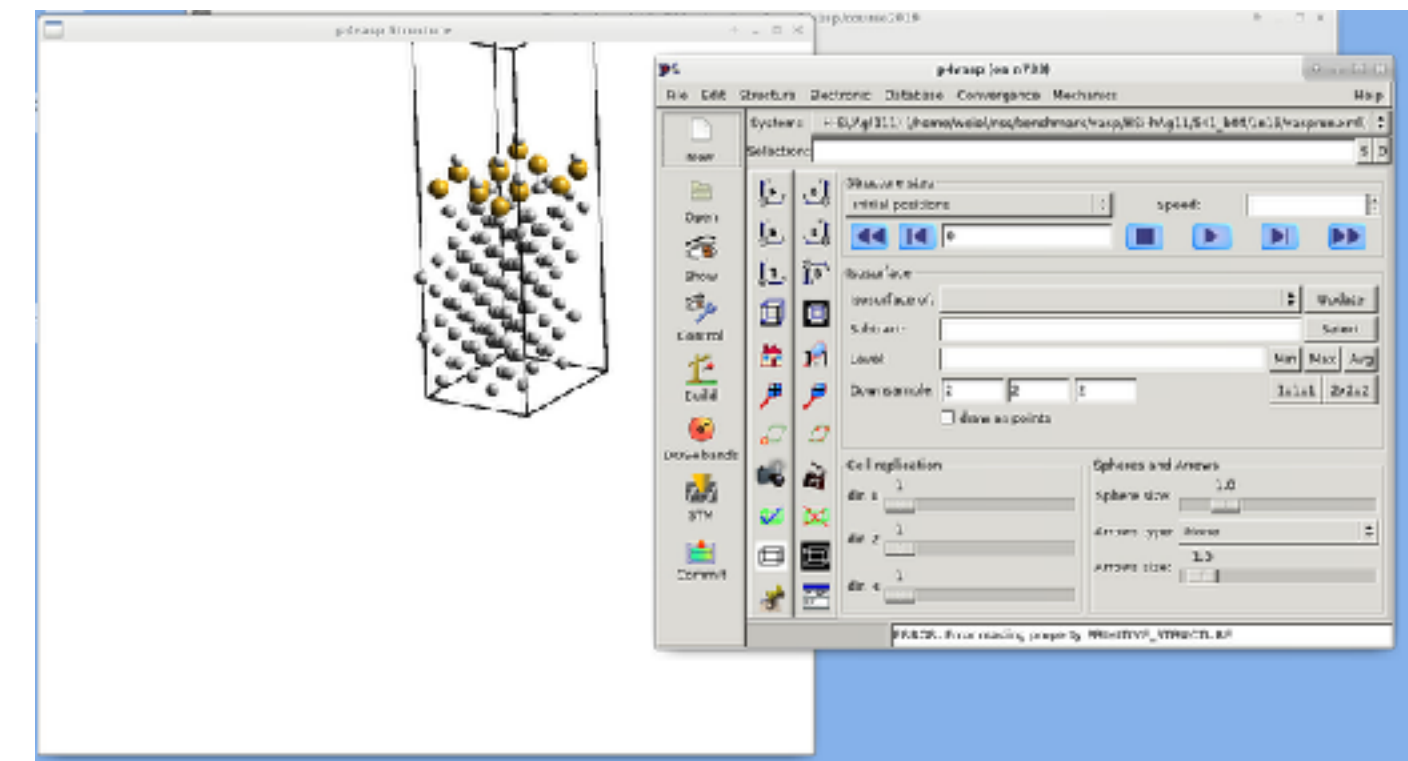
[MaterialsStudio](#)

[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 → 11.000000000000000
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*

- 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`

# 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.
```

-  $\Gamma$ -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
- OpenACC GPU binaries, same names
- + modifications

e.g. constrained relaxation

# 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 - Dardel (PDC)

Example might be outdated!

```
#!/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](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

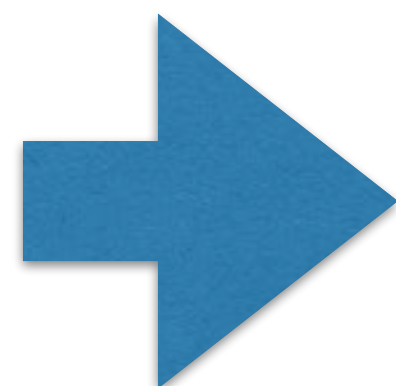
- [DOSCAR](#) - total, partial density of states (DOS)
- [CHGCAR](#) - charge density  
output can also be switched off
- [WAVECAR](#) - plane wave coefficients (for restart)
- ...

# 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   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.)

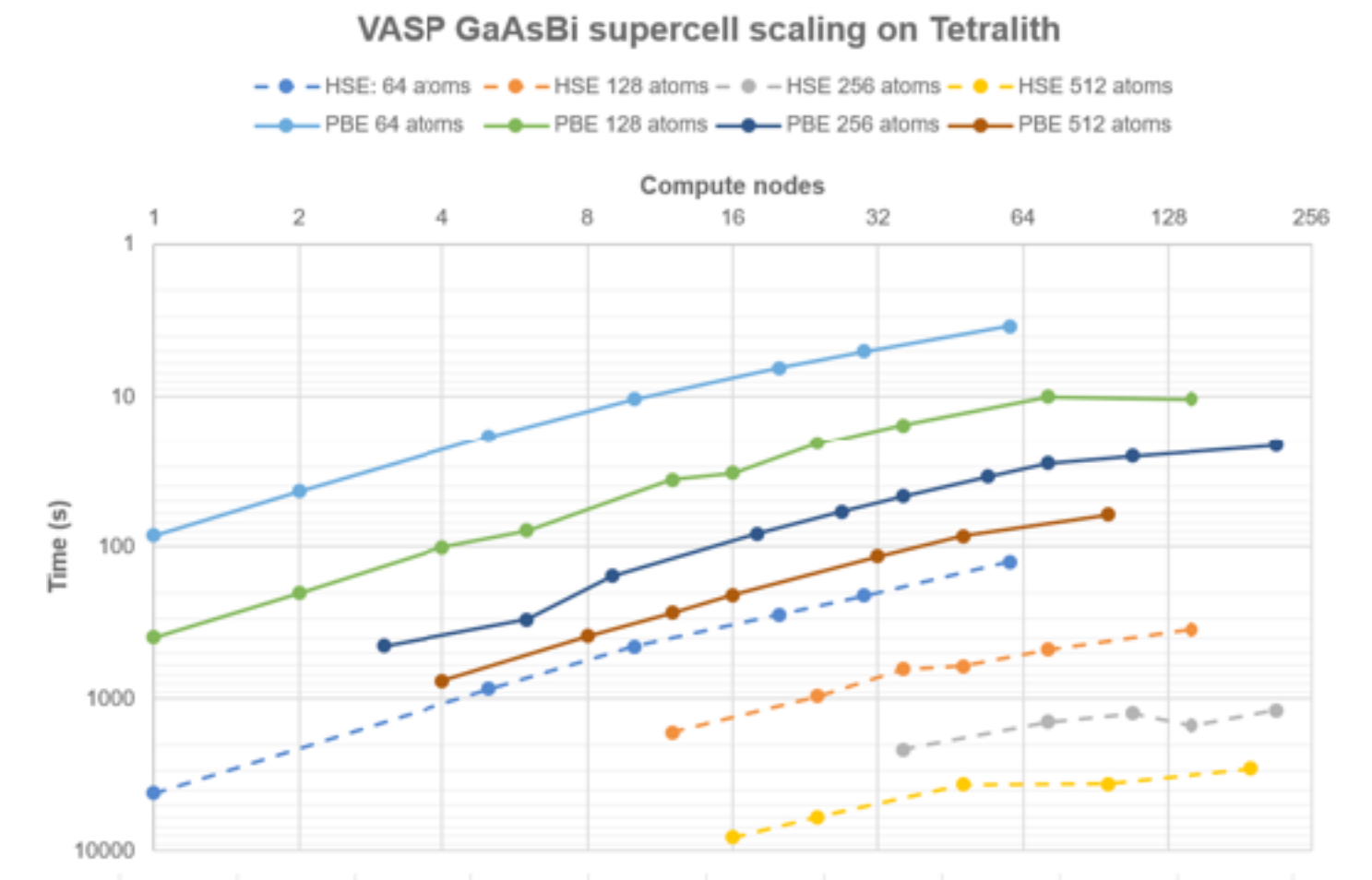
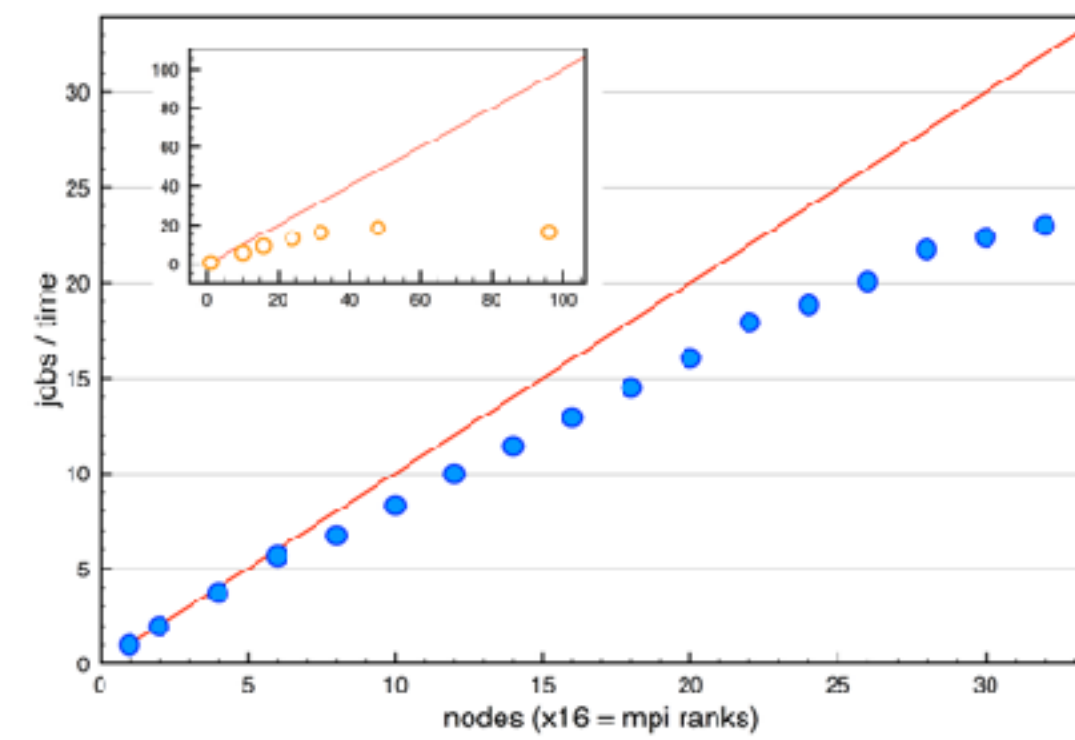
# VASP6

- [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, 6.4 updates -> increased speed (at least 10x noticed)

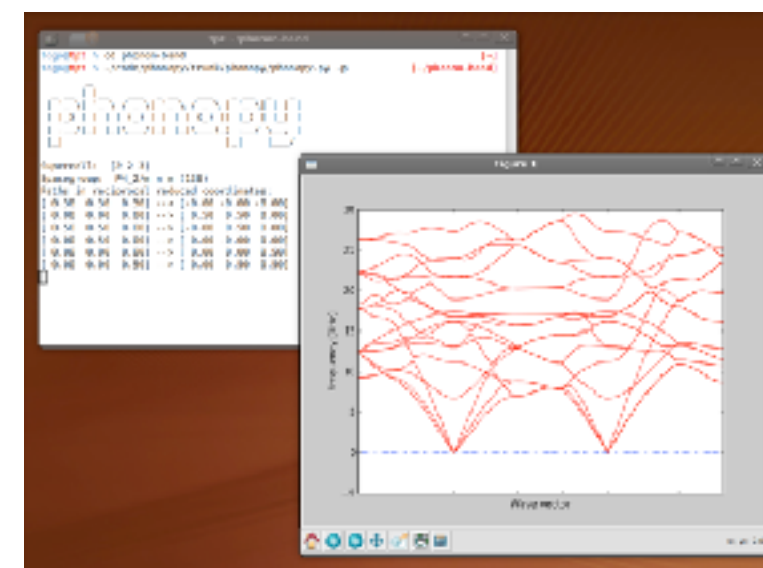
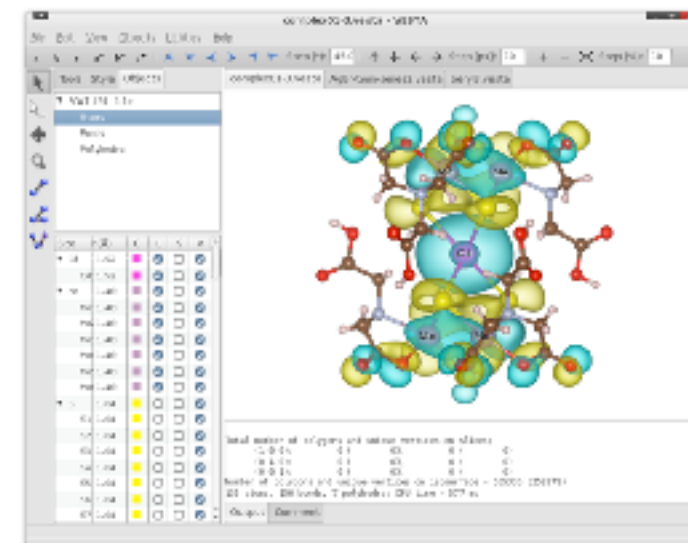
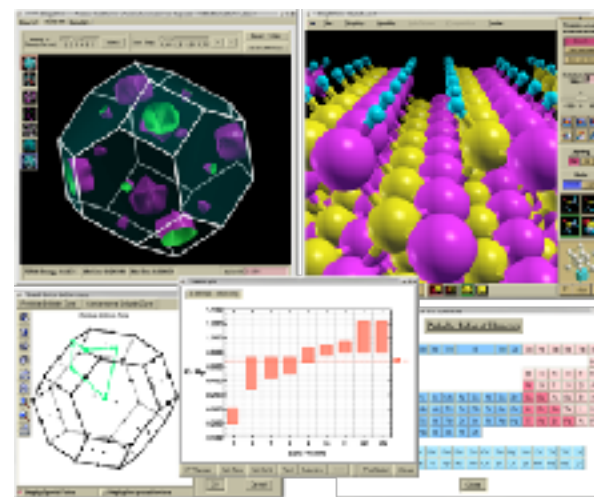
# Tomorrow:

- Running & Performance

- Parallelization
- Efficient settings
- **Problems**



- Utilities & Summary



VTST•Tools

USPEX Computational Materials Discovery