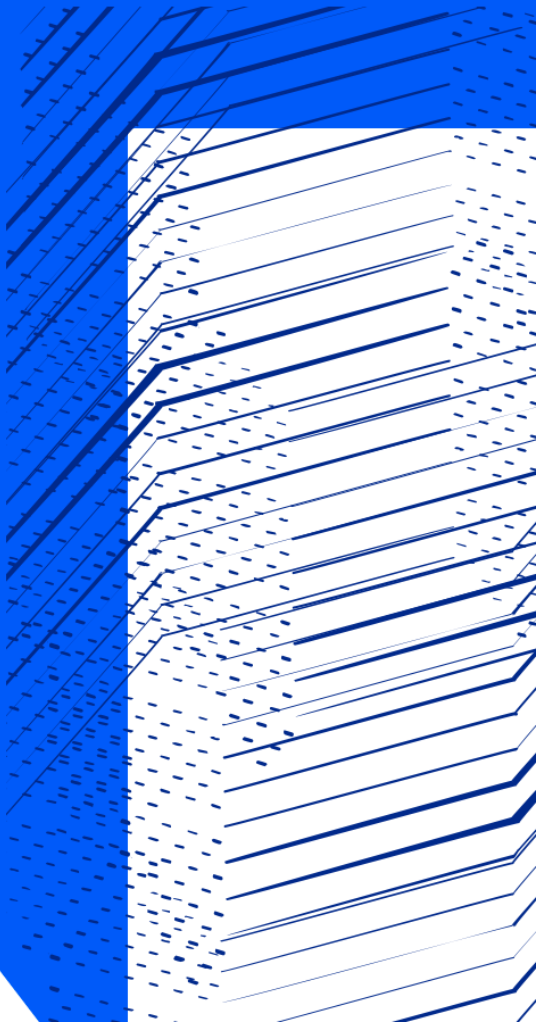
Use all the codes!

Enabling portable researchers

Adam J. Jackson

Email: adam.jackson{at}stfc.ac.uk
Gitlab/Github: @ajjackson Twitter: @binarystate



Overview

1 Background

Why is everything blue?

2 Portable researchers

and their tools

3 Plane-wave DFT in ASE

VASP and CASTEP: similar codes, different interfaces

Science and Technology Facilities Council

About me

- 2006-2011 MEng Chemical Engineering (University of Bath)
- 2011-2016 MRes → PhD → Research Chemistry (University of Bath)
  - *Ab initio* thermodynamics
  - Photovoltaics
- 2016-2019 Post-doc Chemistry (University College London)
  - Semiconductor defects
  - Transparent conductors
- 2019- Computational Scientist Theoretical Physics (STFC)
  - Inelastic neutron scattering

"DFT practitioner"

Science and Technology Facilities Council

Rutherford Appleton Laboratory

STFC is a UK research council with a mission to:

Promote and support high-quality scientific and engineering research by developing and providing, by any means, facilities and technical expertise in support of basic, strategic and applied research programmes funded by persons established in the UK and elsewhere.

i.e. SFTC *does research*

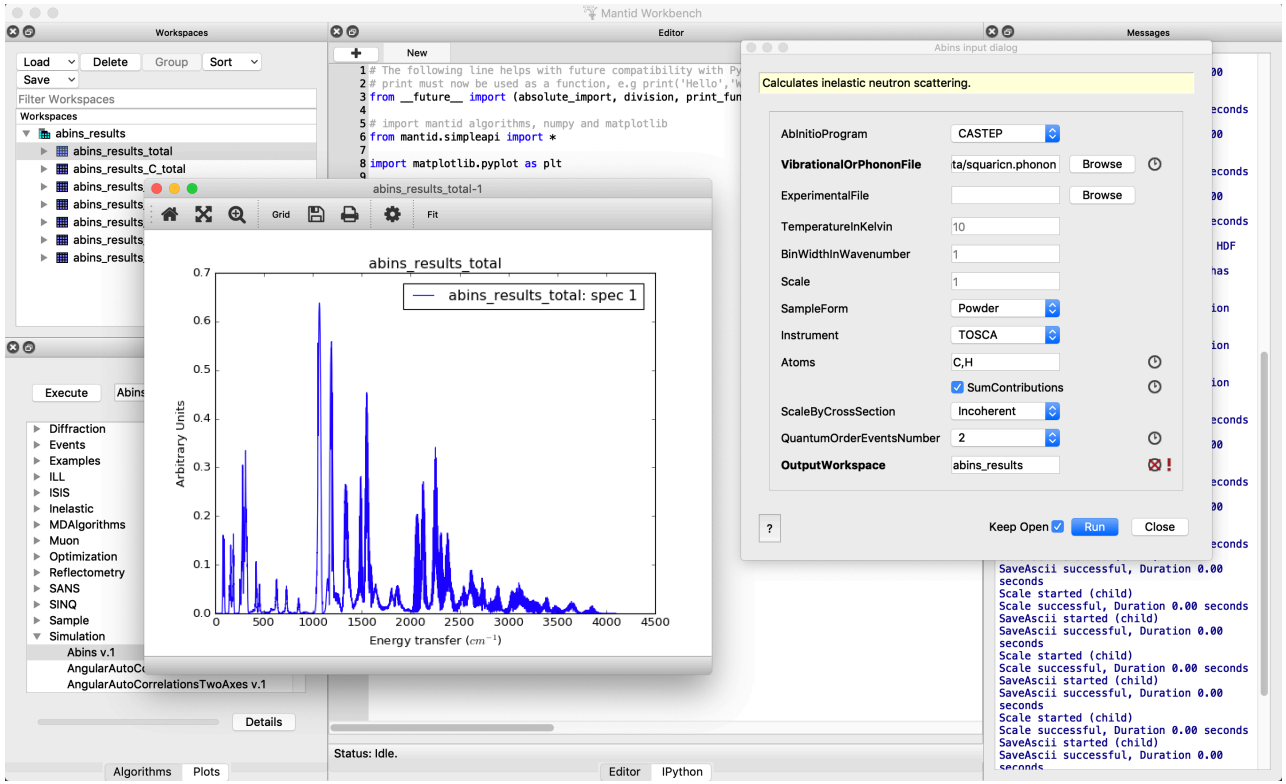
RAL site includes

- computing/data infrastructure,
- Central Laser Facility
- ISIS neutron and muon source
- Diamond light source (joint venture with Wellcome Trust)

Science and Technology Facilities Council

Current work: Abins / Mantid

- Inelastic neutron scattering (INS) simulation
- Abins is an “Algorithm” within Mantid, a framework for instrument scientists
- Currently imports vibration/phonon calculation data from CASTEP, CRYSTAL, Gaussian, DMOL3



Science and Technology Facilities Council

Science and Technology Facilities Council

Portable researchers

So many codes…

Science and Technology Facilities Council

Codes used in an early research career

(sample of 1)

- 2011-2016 (Masters, PhD) *Ab initio thermodynamics, photovoltaics*
  - FHI-aims (DFT, atom-centred numerical orbitals, all-electron)
  - GULP (forcefield)
  - VASP (DFT, plane-wave basis, pseudopotential)
- 2016-19 (Post-doc) *Defects, optics, transparent conductors*
  - VASP
  - GPAW (DFT, numerical basis sets, pseudopotential)
  - Questaal (GW, LMTO, all-electron)
- 2019-? (National lab) *Phonons, INS simulation*
  - CASTEP (DFT, plane-wave basis, pseudopotential)
  - CRYSTAL (DFT, Gaussian basis, all-electron)

How to choose an atomistic code

Good reasons to choose a particular code

- Available levels of theory (forcefields, DFT, GW etc.)
- Available analysis tools
- Performance optimisation for available HPC
- Accuracy/reliability established by benchmarking
- Reproducibility of calculations and long-term sustainability of project

Realistic reasons to choose a particular code

- Your supervisor/colleagues already know it
- Cost/availability of licenses
- Your supervisor/colleagues are friends with the developers
- All of your group's existing tools were written for it

UKRI Science and Technology Facilities Council

How to choose an atomistic code

Good reasons to choose a particular code

- Available levels of theory (forcefields, DFT, GW etc.)
- Available analysis tools
- Performance optimisation for available HPC
- Accuracy/reliability established by benchmarking
- Reproducibility of calculations and long-term sustainability of project

Realistic reasons to choose a particular code

- Your ~~supervisor/colleagues~~ already know it
- Cost/availability of licenses
- Your ~~supervisor/colleagues~~ are friends with the developers
- All of your ~~group's~~ existing tools were written for it

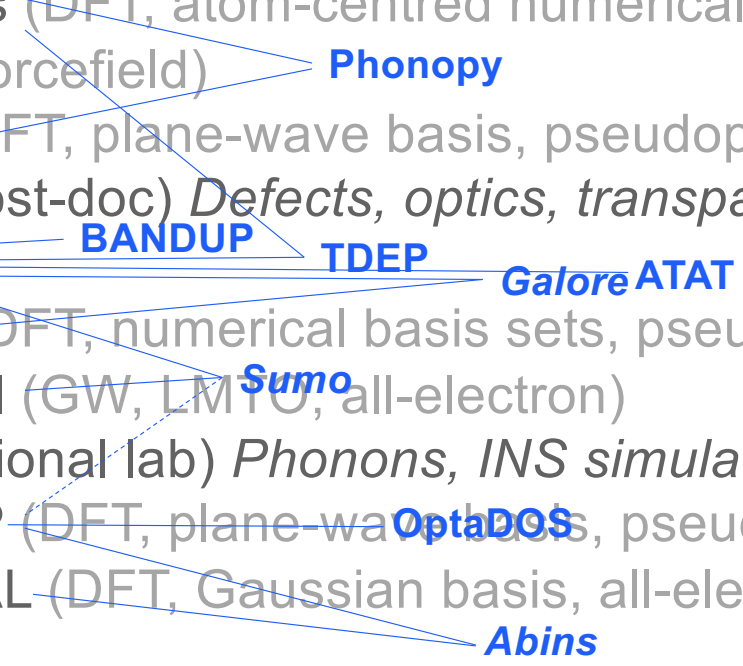
Science and Technology Facilities Council

Becoming independent doesn't actually change this?

Codes used in an early research career

(sample of 1)

- 2011-2016 (Masters, PhD) *Ab initio thermodynamics, photovoltaics*
  - FHI-aims (DFT, atom-centred numerical orbitals, all-electron)
  - GULP (forcefield)
  - VASP (DFT, plane-wave basis, pseudopotential)
- 2016-19 (Post-doc) *Defects, optics, transparent conductors*
  - VASP
  - GPAW (DFT, numerical basis sets, pseudopotential)
  - Questaal (GW, LMTO, all-electron)
- 2019-? (National lab) *Phonons, INS simulation*
  - CASTEP (DFT, plane-wave basis, pseudopotential)
  - CRYSTAL (DFT, Gaussian basis, all-electron)



Science and Technology Facilities Council

Importance of good tools

ASE addresses the “tool compatibility” problems:

- By writing personal/internal tools with ASE, we have a low barrier to supporting more atomistic codes

- By developing community tools and new methods with ASE, we bring them to more researchers

A question for coffee / beer / hackathon: can ASE help with the other factors?

Science and Technology Facilities Council

Importance of good tools

Anecdote 1: Everyone needs file converters

- 2012 – I write a shell script aims2vasp to convert aims structures to VASP input. It's not very good.
- 2014 – I write ase-convert, a wrapper around ASE io functions. With only a few lines of code it's much more useful but needs maintenance.
- 2017 – @ithod adds ase convert command to ASE. It's essentially identical to ase-convert.
- 2019 – ase convert has had 13 more commits for improvements and maintenance.

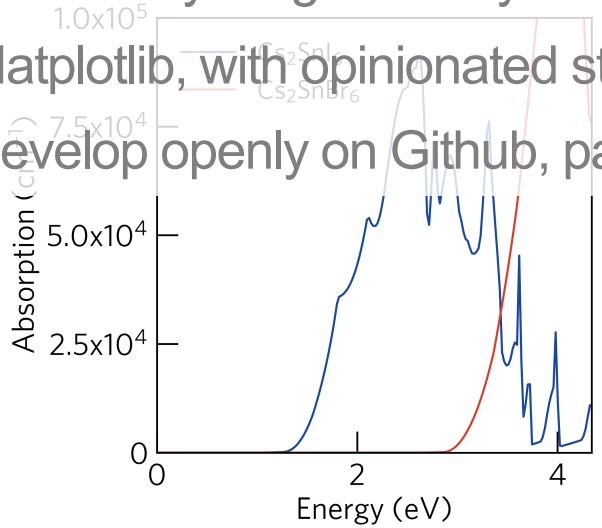
Importance of good tools

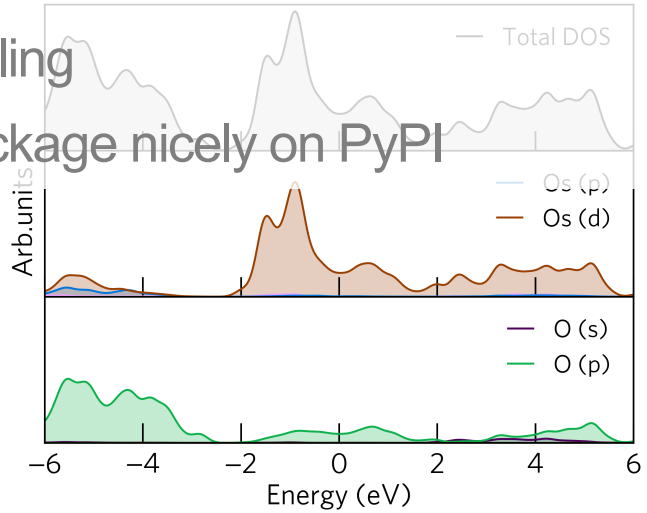
Anecdote 2: Sumo

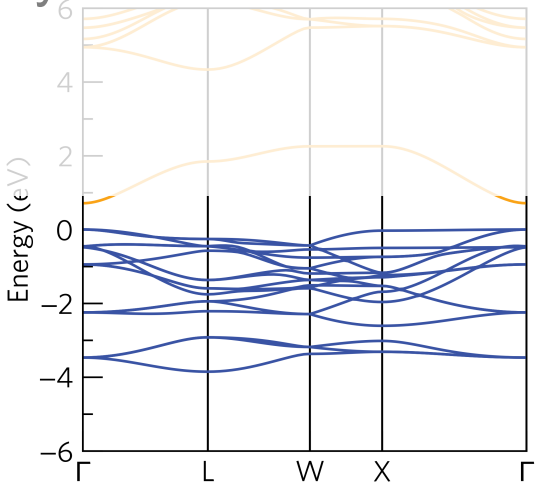
JOSS 10.21105/joss.00717

- Sumo is a package developed in the Scanlon Materials Theory Group at UCL
- Started as “script pile”; tools for setting up and plotting VASP calculations

- Then we made a few strong design decisions:
  - Use the Pymatgen library for Vasp IO, data containers and utility functions
  - Matplotlib, with opinionated styling
  - Develop openly on Github, package nicely on PyPI







Importance of good tools

Anecdote 2: Sumo – the good

- Training time/effort drastically reduced
- Nice student reports, less mistakes
- As people left the group they retain access to their work, including updates
- Support was added for Questaal, and is in-progress for CASTEP
- External users provide bug fixes and ideas for new features

Importance of good tools

Anecdote 2: Sumo – the bad

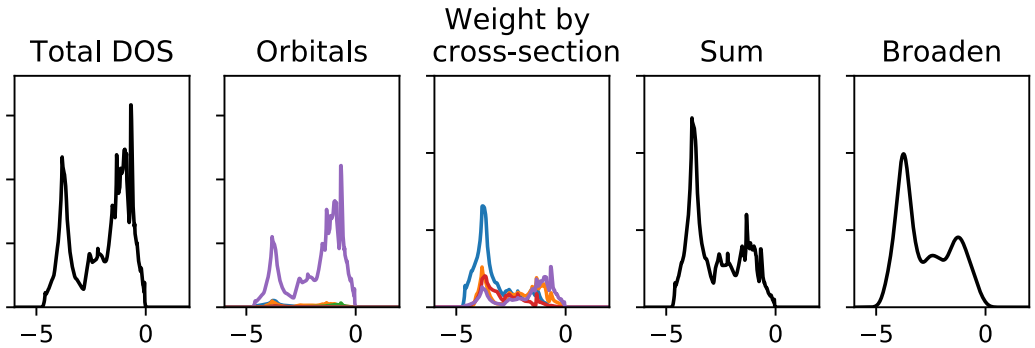
- Pymatgen tends to move fast and break things; not ideal for a dependency.
- Adding support for other codes is cumbersome; partly because Pymatgen is Vasp-centric, and partly because every code has a different workflow!
- Both of the main developers have moved to development-heavy roles elsewhere. Issues and PRs can sit dormant for a while…

Importance of good tools

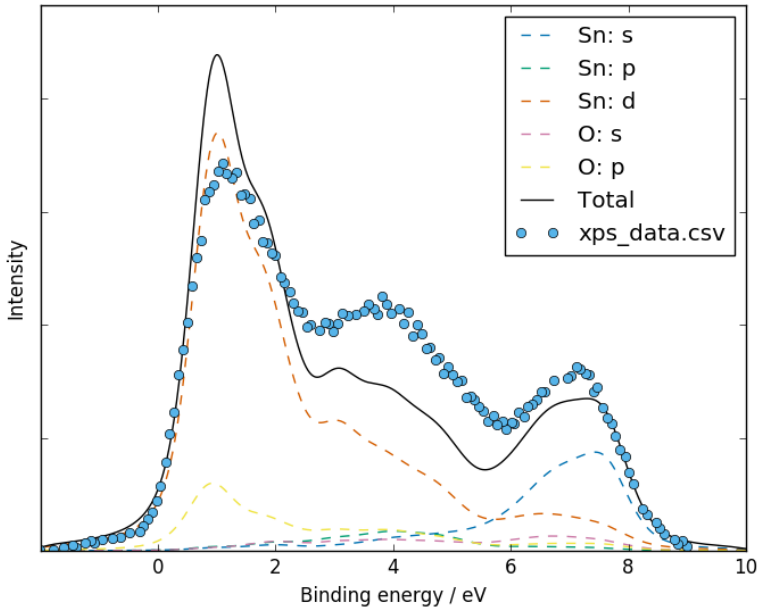
Anecdote 3: Galore

JOSS 10.21105/joss.00773

- Galore is a little package for simulating photoelectron spectra
- It automates some simple but tedious transformations to an orbital-projected DOS
- It was mostly developed while working for Scanlon group at UCL, so the priority was VASP



Science and Technology Facilities Council



Importance of good tools

Anecdote 3: Galore

- In Feb 2018 there was a pensions dispute and many university workers went on strike
- One challenge for striking academic researchers is that their future career depends on their *outputs*
- "What can I do today that won't help UCL but furthers my selfish research interests?"
- Adding support for GPAW was *really easy*

Commits on Mar 2, 2018

Paper: AR Imperial acknowledgement
ajjackson committed on 2 Mar 2018

Paper: some wording tweaks, refer to figure 3.
ajjackson committed on 2 Mar 2018

Merge branch 'master' into paper
ajjackson committed on 2 Mar 2018

Commits on Feb 26, 2018

ENH: Import TDOS or PDOS from GPAW calculations
ajjackson committed on 26 Feb 2018

Commits on Feb 19, 2018

DOC: update HAXPES tutorial to use Scofield fit
ajjackson committed on 19 Feb 2018

REL: Version stamps for 0.3.0 release
ajjackson committed on 19 Feb 2018

ENH: Rename Yeh data sets from XPS, UPS to Alka, He2
ajjackson committed on 19 Feb 2018

DOC: README updates for PES emphasis and galore-get-cs
ajjackson committed on 19 Feb 2018

Science and Technology Facilities Council

Conclusions

- Researchers can be brought into contact with many different atomistic codes – not always for obvious reasons!
- Analysis packages enable new and different science, while simple utilities can dramatically affect productivity
- Open-source development ensures access and rights to benefit from your previous work
- Technical compatibility is also needed if we are to take advantage of this
- Some codes are easier to interface than others

Science and Technology Facilities Council

ASE, Vasp and CASTEP

So close, yet so far…

Science and Technology Facilities Council

Two plane-wave pseudopotential codes

VASP	CASTEP
Development led in Vienna, Austria	Development led in Cambridge, UK
Fortran, MPI	Fortran, MPI
Somewhat expensive	Free (for non-commercial use) Pretty expensive (for commercial use)
Commercial framework MedeA	Commercial framework Materials Studio
Highlights: post-HF (RPA, MP2, BSE), meta-GGA, library of trusted PPs, magnetism, FFT parallelism, TD-HF	Highlights: DFPT, NMR shifts, EELS, on-the-fly PPs, inbuilt help and unit conversions, preconditioned geometry optimization, efficient phonon methods
Interfaces: ASE, Phonopy, ShengBTE, Pymatgen, TDEP, ATAT, BANDUP, Calypso, USPEX, AIRSS, KLMC, OCLIMAX, *Sumo, Galore*…	Interfaces: ASE, ShengBTE, OptaDOS, Calypso, USPEX, AIRSS, *Abins,* OCLIMAX, Soprano, …

Two plane-wave pseudopotential codes

VASP	CASTEP
Input files: fixed names `POSCAR`, `POTCAR`, `KPOINTS`, `INCAR`	Input files: two files, "seedname" with extensions `seedname.cell`, `seedname.param`
Output files: binary (e.g. `WAVECAR`), text (e.g. `DOSCAR`) and XML (`vasprun.xml`)	Output files: binary (e.g. `seedname.pdos_weights`) and text (e.g. `seedname.phonon`)
Esoteric options: EDIFFG sets an energy criterion if positive and a force criterion if negative	Forgiving input: e.g. KPOINTS_MP_GRID is aliased to KPOINT_MP_GRID
Documentation: online wiki	Documentation: Inbuilt help
Workflow: Tasks such as band structures and RPA can involve multiple runs of VASP with different input and precalculated density, orbitals. Limited restart ability.	Workflow: Band structure and optics "tasks" allow separate k-point meshes for SCF and properties. Powerful "phonon" task can restart unfinished DFPT calc.
Missing batteries: No plotting included (*Sumo*). External tools needed for off-Gamma phonons (*Phonopy*).	Missing batteries: *OptaDOS* is required for "basic" PDOS, optics analysis.

Nice features in ASE Calculators

VASP

- The Vasp calculator provides “recipes” for methods which would usually require several flags to be set.
- These have less confusing names (‘revpbe’ -> ‘RE’)
- Per-atom properties (Hubbard U and initial magnetic moments) have awkward INCAR syntax. ASE takes care of this for you.
- Structure is transparently sorted/unsorted to reduce file size/complexity

```
        'rpbe': {'gga': 'RP'},
        'am05': {'gga': 'AM'},
        # Meta-GGAs
        'tpss': {'metagga': 'TPSS'},
        'revtpss': {'metagga': 'RTPSS'},
        'm06l': {'metagga': 'M06L'},
        'ms1': {'metagga': 'MS1'},
        'ms2': {'metagga': 'MS2'},
        'scan': {'metagga': 'SCAN'},
        # vdW-DFs
        'vdw-df': {'gga': 'RE', 'luse_vdw': True, 'aggac': 0.},
        'optpbe-vdw': {'gga': 'OR', 'luse_vdw': True, 'aggac': 0.0},
        'optb88-vdw': {'gga': 'BO', 'luse_vdw': True, 'aggac': 0.0,
                       'param1': 1.1 / 6.0, 'param2': 0.22},
        'optb86b-vdw': {'gga': 'MK', 'luse_vdw': True, 'aggac': 0.0,
                        'param1': 0.1234, 'param2': 1.0},
        'vdw-df2': {'gga': 'ML', 'luse_vdw': True, 'aggac': 0.0,
                    'zab_vdw': -1.8867},
        'rev-vdw-df2': {'gga': 'MK', 'luse_vdw': True, 'param1': 0.1234,
                        'param2':0.711357, 'zab_vdw': -1.8867, 'aggac': 0.0},
        'beef-vdw': {'gga': 'BF', 'luse_vdw': True,
                     'zab_vdw': -1.8867},
        # Hartree-Fock and hybrids
        'hf': {'lhfcalc': True, 'aexx': 1.0, 'aldac': 0.0,
               'aggac': 0.0},
```

Science and Technology Facilities Council

Nice features in ASE Calculators

CASTEP

- The inbuilt help system is automatically read to collect information about valid keys *for your build of CASTEP*. This allows **tab completion.**
- Dict-style k-point settings have been implemented, providing access to Gamma-centered meshes – more convenient than working out offsets.
- CASTEP is case-insensitive… and so is its ASE property interface.

```
[6]: calc = Castep(xc='pbe', kpts={'size': (4, 4, 4), 'gamma': True})

[7]: calc.param.phonon_
```

```
phonon_calc_lo_to_splitting     phonon_dos_spacing              phonon_force
phonon_calculate_dos            phonon_energy_tol               phonon_force
phonon_const_basis              phonon_fine_cutoff_method       phonon_max_c
phonon_convergence_win          phonon_fine_method              phonon_max_c
phonon_dfpt_method              phonon_finite_disp              phonon_metho
phonon_dos_limit                phonon_force_constant_cut_scale phonon_preco
```

Challenges for ASE Calculators

VASP

- Many keywords undocumented or part of community mods/extensions
- Sorting/unsorting is complicated, leads to surprises for users – edge cases
- Standard output file easily corrupted, overflows fields
- XML file is invalid if run killed while a tag is open
- Multiple competing implementations within ASE project

```
'external_pressure',  # Pressure for NPT calcs., equivalent to
'lj_radius',   # Undocumented classical vdW parameter
'lj_sigma',    # Undocumented classical vdW parameter
'mbd_beta',    # TS MBD vdW correction damping parameter
'scsrad',      # Cutoff radius for dipole-dipole interaction ter
'hitoler',     # Iterative Hirschfeld partitioning tolerance
'lambda',      # "Spring constant" for magmom constraint calcs.
'kproj_threshold',  # Threshold for k-point projection scheme
'maxpwamp',    # Undocumented HF parameter
'vcutoff',     # Undocumented parameter
'mdtemp',      # Temperature for AIMD
'mdgamma',     # Undocumented AIMD parameter
'mdalpha',     # Undocumented AIMD parameter
'ofield_kappa',  # Bias potential strength for interface pinnir
```

Science and Technology Facilities Council

Challenges for ASE Calculators

CASTEP

- The main output files are a “human-readable” *.castep* file and “machine-readable” *.castep_bin* file
- Currently we read the *.castep file* – sensitive to verbosity, missing data
- The *.castep_bin* file **should** be more reliable, but…
  - Formatted binary from Fortran
  - No formal specification, version-dependant
  - In discussion with CASTEP developers to make a Python parsing library

Challenges for ASE Calculators

CASTEP

- Parameter system is complicated
  - The *seedname.cell* + *seedname.param* file does not fit ASE's Atoms + Calculator model – settings must be dispatched to the correct file
  - Parameter setting is not consistent with other ASE Calculators
- Reading *too much* information from output – explicit defaults, redundancy
- Incompatible parameters require logic to "clobber" existing keys

Observations

- CASTEP generally has a superior user interface
- Several features that are helpful to humans (aliasing, auto verbosity, file distribution, units) *make it harder* to wrap in ASE

- VASP has a poor user interface, but is generally simpler to wrap due to
  - More predictable file contents
  - Limited formatting options within a single file
  - Use of machine-readable XML

- In both cases we have used the ASE Calculator to provide additional convenience tools.
- Vasp interface development is more active – but coordination is an issue

Acknowledgements

- Thanks to the conference organisers
- Other people did most of the work on the VASP and CASTEP calculators

- Alex Ganose (Sumo lead, now LBNL)

- STFC colleagues
  - Dominik Jochym (my boss – email dominik.jochym{at}stfc.ac.uk for academic CASTEP licenses. One email per group…)
  - Simone Sturniolo (CASTEP calculator, Python/binary interface)

- CoSeC / UKCP (supporting this trip)

Science and Technology Facilities Council

Thank you

Science and Technology Facilities Council @STFC_Matters Science and Technology Facilities Council