Story of ASE

ASE Workshop
Chalmers, November 19-22, 2019

Jens Jørgen Mortensen, *CAMd, Department of Physics, Technical University of Denmark*

- ASE-1.0
- ASE-2.0
- ASE-3.0
- The future

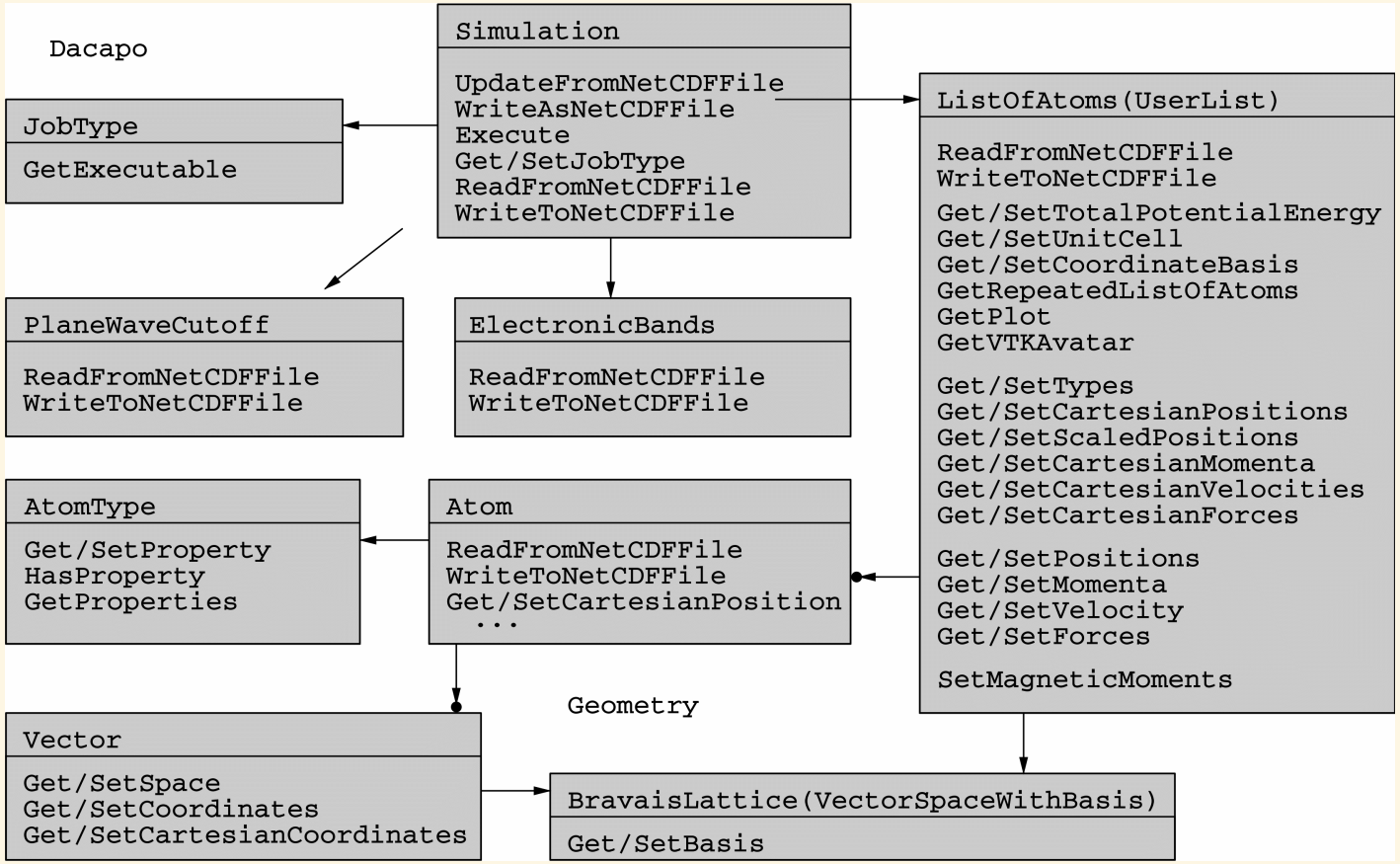
ASE-1.0

It all started with Dacapo (a Fortran USPP plane-wave DFT code):

S. R. Bahn and K. W. Jacobsen, *An object-oriented scripting interface to a legacy electronic structure code*, Comput. Sci. Eng., Vol. **4**, 56-66, 2002, https://dx.doi.org/10.1109/5992.998641

- Use Python scripts
- Write input to netCDF file
- Communicate with Fortran code through a socket
- Read output from netCDF file

Overview


Dacapo
Simulation
UpdateFromNetCDFFile
WriteAsNetCDFFile
Execute
Get/SetJobType
ReadFromNetCDFFile
WriteToNetCDFFile
JobType
GetExecutable
ListOfAtoms(UserList)
ReadFromNetCDFFile
WriteToNetCDFFile
Get/SetTotalPotentialEnergy
Get/SetUnitCell
Get/SetCoordinateBasis
GetRepeatedListOfAtoms
GetPlot
GetVTKAvatar
Get/SetTypes
Get/SetCartesianPositions
Get/SetScaledPositions
Get/SetCartesianMomenta
Get/SetCartesianVelocities
Get/SetCartesianForces
Get/SetPositions
Get/SetMomenta
Get/SetVelocity
Get/SetForces
SetMagneticMoments
PlaneWaveCutoff
ReadFromNetCDFFile
WriteToNetCDFFile
ElectronicBands
ReadFromNetCDFFile
WriteToNetCDFFile
AtomType
Get/SetProperty
HasProperty
GetProperties
Atom
ReadFromNetCDFFile
WriteToNetCDFFile
Get/SetCartesianPosition
...
Geometry
Vector
Get/SetSpace
Get/SetCoordinates
Get/SetCartesianCoordinates
BravaisLattice(VectorSpaceWithBasis)
Get/SetBasis

Example

```
from Simulations.Dacapo import *
mysim = Simulation()
mysim.bands = ElectronicBands(9)
mysim.config = ListOfAtoms(
    atoms=[Atom(Mg_GGA, Vector([0, 0, 0]))],
    unitcell=BravaisLattice(
        [[-1.425, 1.425, 1.425],
         [1.425, -1.425, 1.425],
         [1.425, 1.425, -1.425]])
mysim.plancut = PlaneWaveCutoff(340)
mysim.Execute()
```

ASE-2.0

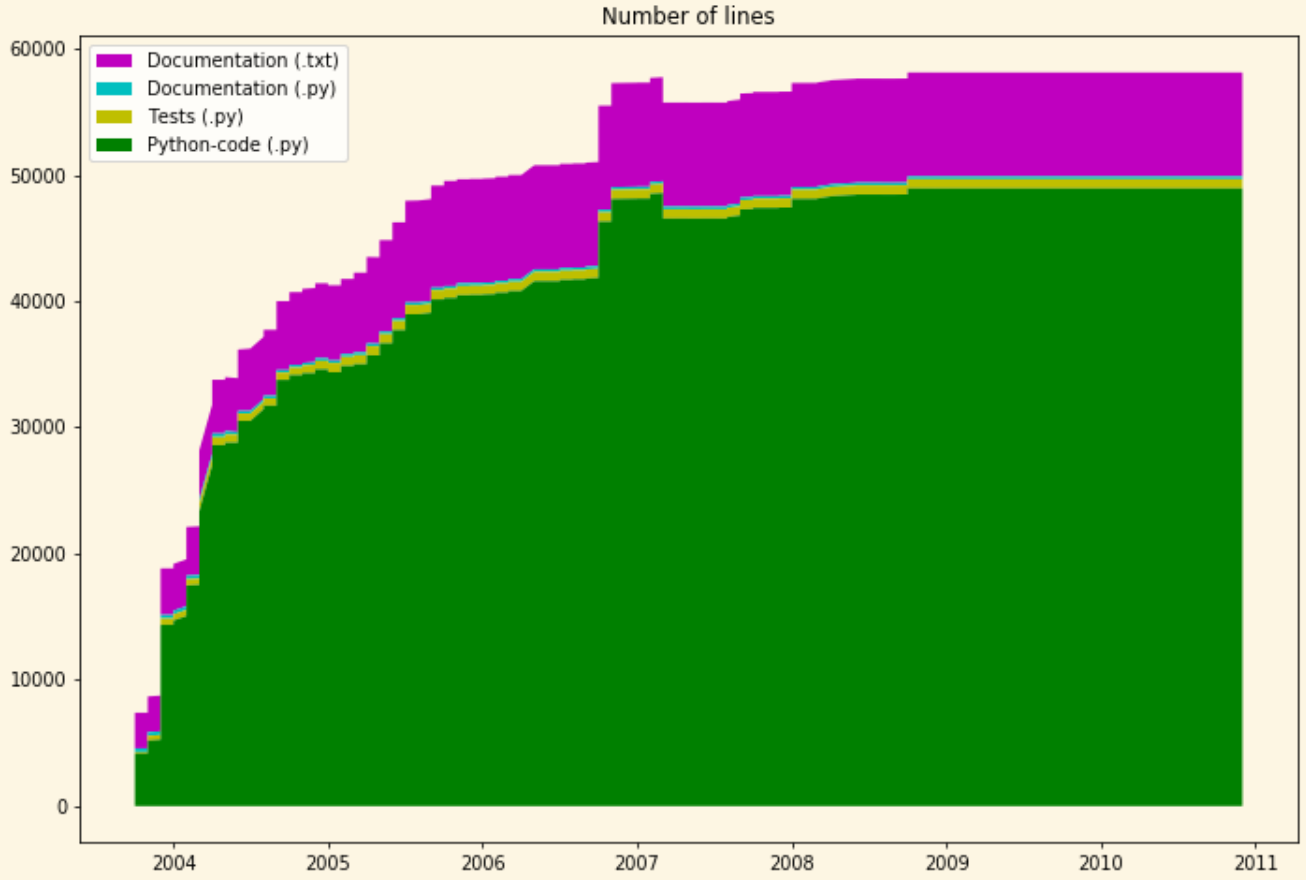
Problems with ASE-1:

- Other codes (from Dacapo to ASAP)
- Millions of atoms
- Too many objects?

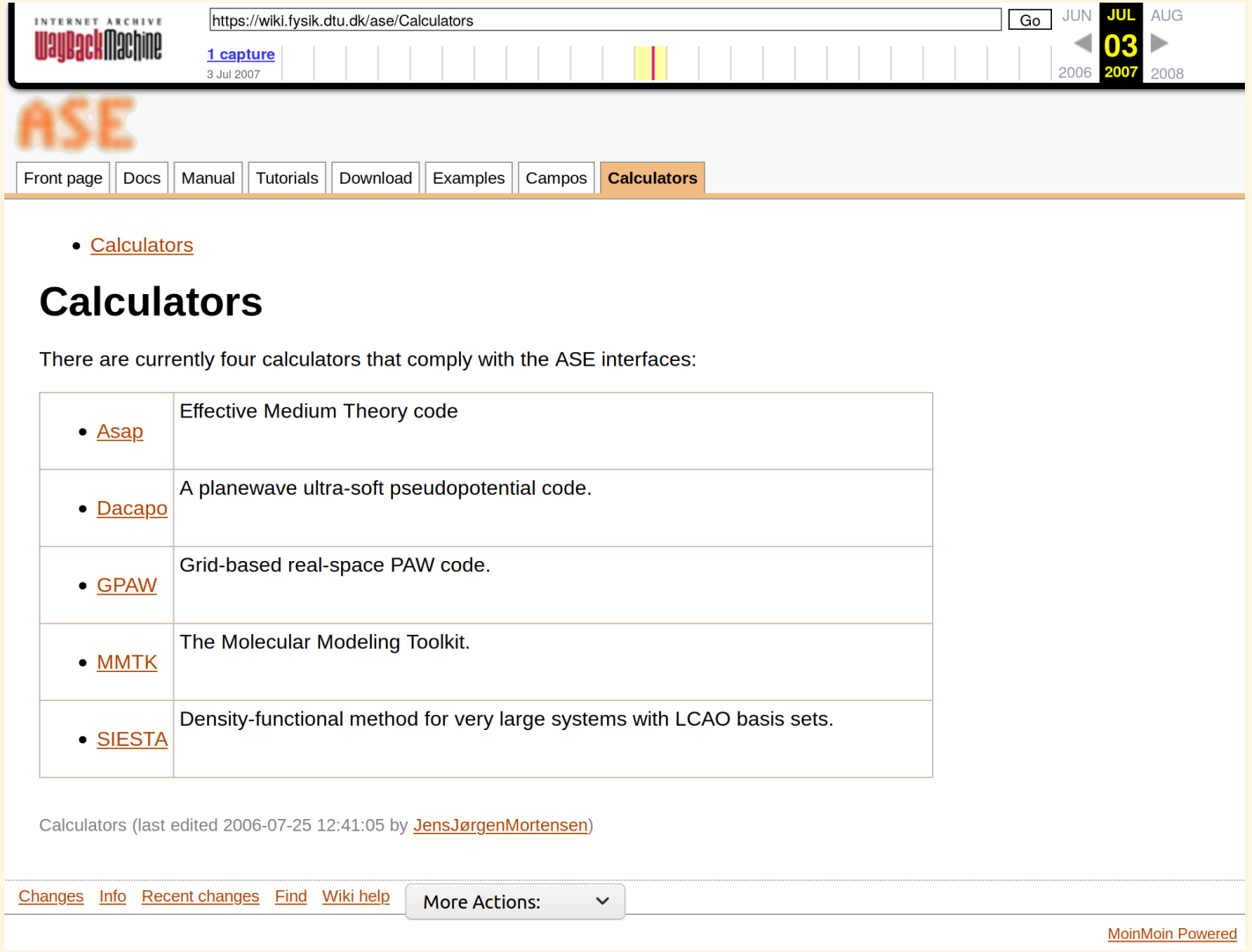
After many discussions and meetings, ASE-2.0 was created from scratch.

- `Simulation` -> `ListOfAtoms`
- `ListOfAtoms.GetCartesianPositions()` -> `numpy.ndarray`

Lines of code



Calculators



ASE-3.19: ~40 calculators

Installation ASE in the old days

Download and install ASE

Contents

Requirements for ASE

The following packages are required for basic ASE functionality:

1. **Python 2.2** or later is required. Python is available from http://www.python.org.
2. **Numeric Python**. Use the latest version 24.2.
   Build preferably from our python-numeric-24.2-1.1.el5.fys.src.rpm or edit **customize.py** file to configure **Numeric Python** to use
3. **Scientific Python** from http://dirac.cnrs-orleans.fr/ScientificPython/. Use the 2.4.11 version.

The following packages are optional, since they enable the use of some specific external tools. You only need to install these tools if

4. **Rasmol** from http://www.openrasmol.org/.
   A rasmol src and binary rpm can be use for installing Rasmol on linux. You can run ASE without RasMol, you just won't be able
5. **Gnuplot**. ASE can use *Gnuplot* via the Gnuplot-Python interface from http://sourceforge.net/projects/gnuplot-py/. At CAMP/Niflh
6. **Python Pexpect** from http://pexpect.sourceforge.net/.
   The current version of *Pexpect* is 2.1, but at Niflheim we use an old version 0.999. Both versions can be downloaded from the a
   RPM packages for *Pexpect* can be found on rpmfind.
7. **Matplotlib** http://matplotlib.sourceforge.net/. Use the 0.90.0 version.
8. **vtk** http://www.vtk.org/. After installation you may need (if **libvtkRenderingPythonTkWidgets.so** is missing) to make a link:

```
ln -s /usr/lib/libvtkRenderingPythonTkWidgets.so.5.0 /usr/lib/libvtkRenderingPythonTkWidg
```

and set the path manually:

```
export VTK_TK_WIDGET_PATH=/usr/lib
```

Use the 5.0.3 version.

ASE-3.0

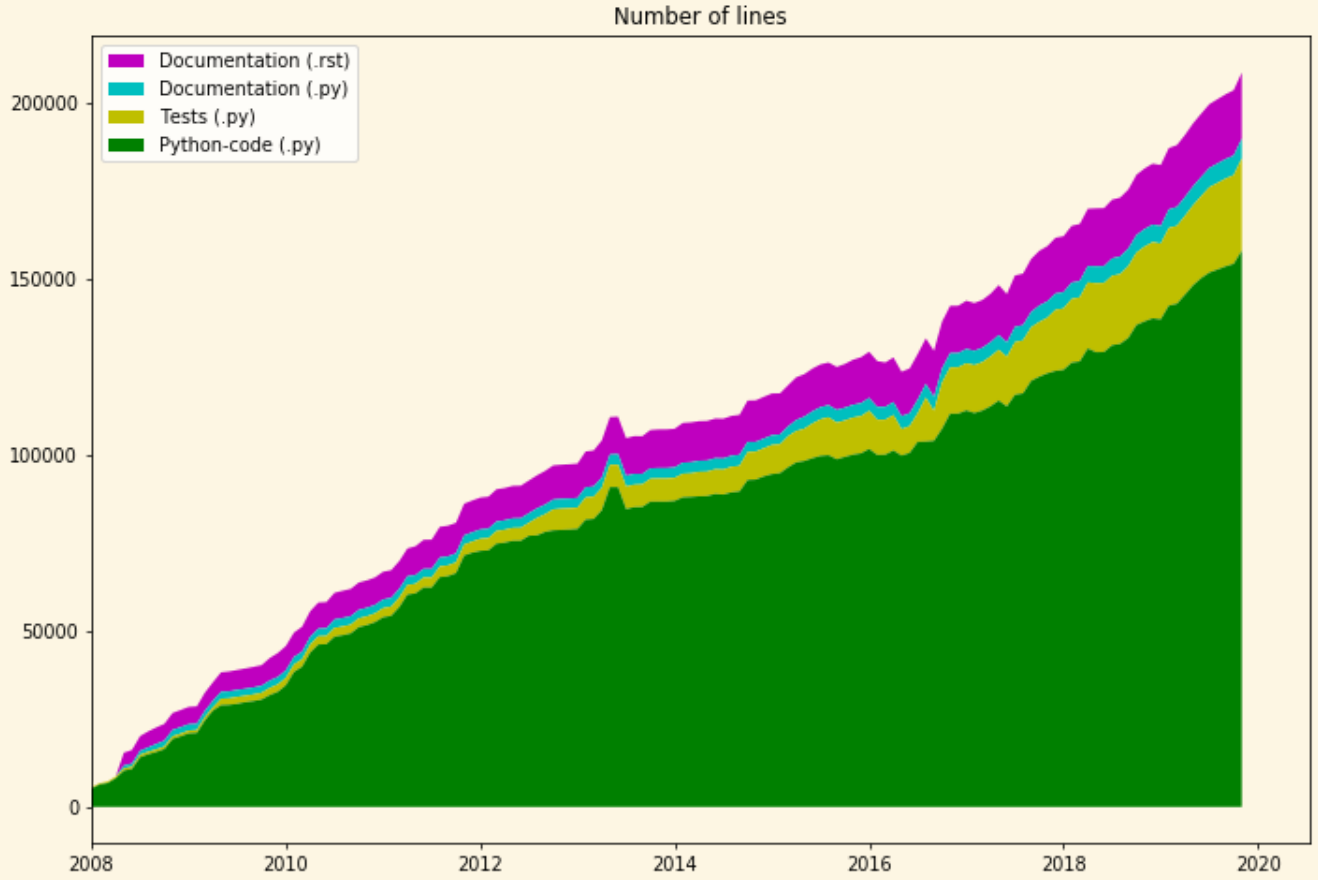
Problems with ASE-2:

- Weird unit system
- Weird naming convention (files and function names)
- `ListOfAtoms` -> `Atoms`
- ...

Rewrite from scratch or adapt current code base?

1. Rewrite
2. Ask for permission later

Lines of code ...



279 contributors so far ...

Some highlights

- A beta version of the new ASE-3.0 will be used for the electronic structure course at CAMd. (10 Jan 2008)
- The new Sphinx page is now up and running! (2 Apr 2008)
- CAMd/Cinf will do a "doc-sprint" from 9 to 16. (17 Apr 2008)
- ASE version 3.0.0 released (13 November 2008).
- ASE has reached revision 1000 (16 July 2009).
- The source code is now on GitLab (18 September 2015).
- Reference paper in J. Phys. Condens. Matter: The Atomic Simulation Environment | A Python library for working with atoms (7 June 2017).

Technology

- Web-page: Home-made thing -> MoinMoin wiki -> Sphinx + Python scripts.
  - https://wiki.fysik.dtu.dk/ase/ -> https://a.se/ ?
- Testing: home-made framework:
  - After every commit: GitLab-CI
  - Every night: VASP + GPAW
- `Numeric` -> `numarray` -> `numpy`, `scipy`
- Python 1.x -> 3.8
- CVS -> SVN -> Git
- `spglib`?, C-code?

You want to contribute to ASE

Thank you!

Let's do it right:

- code review
- documentation
  - low level: docstrings
  - high level: how to use it
- tests
- release notes
- remove old way of doing it

Contribute to ASE?

```
Think of a contribution like a puppy: you might view it as this cute,
wonderful thing you're giving me while I'm looking at it as over a
decade of feeding, walking, and vet bills.

-- Brett Cannon
```

Not everything is a good fit for ASE

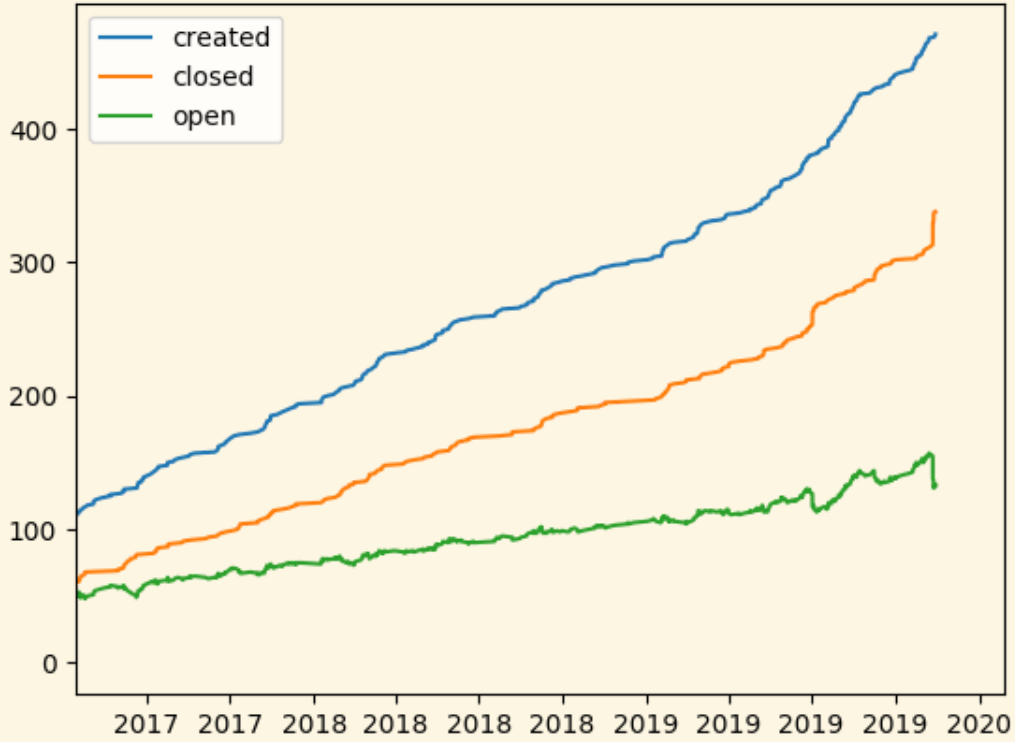
Alternatives:

- Code snippets in a tutorial plus a few helper functions added to ASE.
- Its own thing: PyPI + GitLab + ReadTheDocs + JOSS

The Zen of ASE

- less is more
- so nothing at all could be a good solution
- if your thing has documentation and tests then it *might* be a good fit for ASE
- if it doesn't, we don't want it
- … unless you are from Denmark

Issues and MR's



Merge requests: 1245 merged, 195 closed and 47 open.

Old issues

https://xkcd.com/1822/

http://trac.fysik.dtu.dk/projects/ase

Maintenance

Keeping ASE going is more work that most people think!

- Review and merge MR's
- Fix issues
- Mail-list support
- Find and nudge the expert that knows something

There are many bigger projects necessary in order to keep ASE moving in the right direction. Example: *Calculators*!

Should we try to get funding for an ASE maintainer?

"If you have found ASE to be useful in your work, research or company, please consider making a donation to the project commensurate with your resources. Any amount helps! All donations will be used strictly to fund the development of ASE's open source software, documentation and community."

What could we offer sponsors?

- ASE will also be in good shape two years from now
- Generate goodwill thanks to your appearance in the list of sponsors
- Influence the work of sponsored developers so that ASE continues to fit your needs
- Your favorite electronic structure code will be supported and part of the daily tests.

Type hints

A simple example from ASE's code:

```
import re

def strip_number(s):
    """Split string to number and the rest.

    >>> strip_number('2Au')
    2
    """
    m = re.match('[0-9]*', s)
    return int(m.group() or 1), s[m.end():]

x = strip_number('12Cu') * 2.5
```

See also: https://gitlab.com/ase/ase-workshop-discussion/issues/2

Type hints + MyPy

```python
import re
from typing import Tuple

def strip_number(s: str) -> Tuple[int, str]:
    m = re.match('[0-9]*', s)
    return int(m.group() or 1), s[m.end():]

x = strip_number('12Cu') * 2.5
```

```
$ mypy hint.py
hint.py:6: error: Item "None" of "Optional[Match[str]]"
           has no attribute "group"
hint.py:6: error: Item "None" of "Optional[Match[str]]"
           has no attribute "end"
hint.py:8: error: Unsupported operand types for * ("Tuple[int, str]"
                                                   and "float")
Found 3 errors in 1 file (checked 1 source file)
```

Type hints + MyPy ...

```
import re
from typing import Tuple

def strip_number(s: str) -> Tuple[int, str]:
    m = re.match('[0-9]*', s)
    assert m is not None
    return int(m.group() or 1), s[m.end():]

x = strip_number('12Cu')[0] * 2.5
```

```
$ mypy hint.py
Success: no issues found in 1 source file
```

Type hint in ASE?

- we can do it gradually
- it's documentation that mypy can check for us
- code is read a lot more than written
- make it easier for new contributors
- improved editor-completion
- find bugs for us
- lead to simpler code
- wait for ASE to drop Python 3.5:

```
...
number = []  # type: List[int]
number: List[int] = []
```

Summary

- ASE has been rewritten from scratch twice - I hope we will not need a 4.0 rewrite
- Please donate to ASE - somehow
- Type hints are great!