Introduction to Scripting

Thomas M. Soini
Ole Carstensen
Hans van Schoot

TCCM Workshop | April 22 | Amsterdam

Scripting – Why?

- Individual calculations:
  ADF-GUI or write .run file

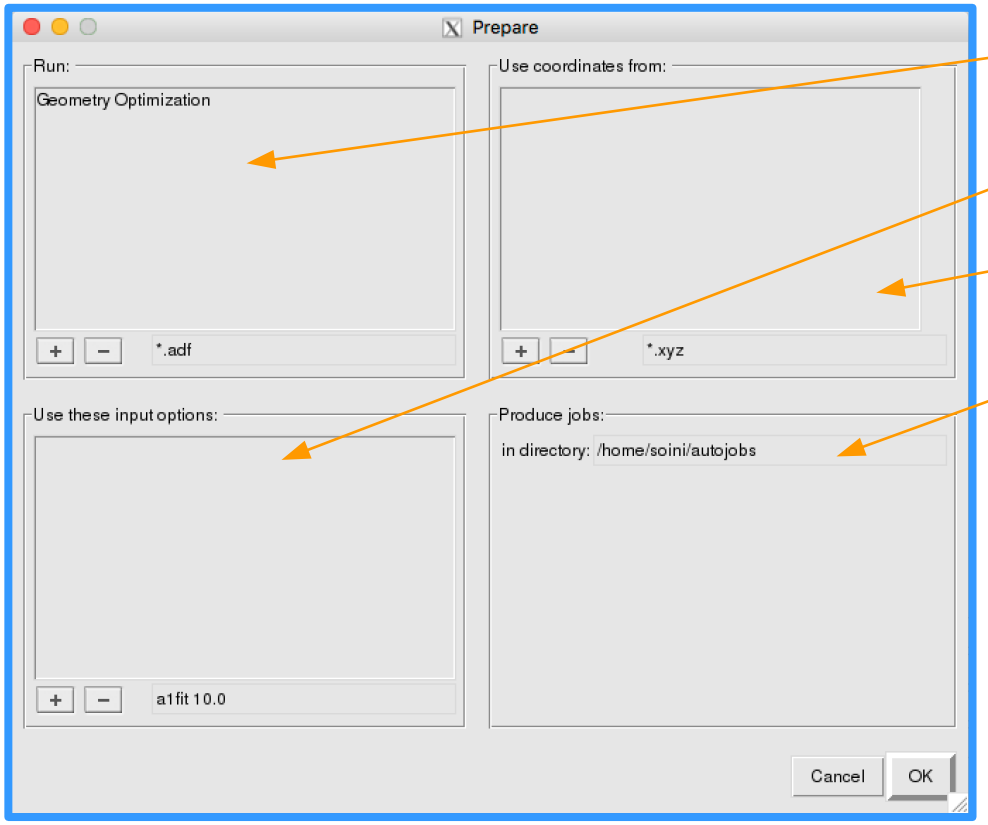
- Modern computational chemistry and materials science may involve
  - lots of calculations
  - repetitive tasks
  - complex workflows

- Scripting in Quantum Chemistry
  - increase productivity
  - reduce human errors

SCM

Series of ADF Calculations – Non-Scripted Case

- Task: prepare multiple ADF jobs (eventually very many!)
- Can be accomplished within ADF-GUI
  - Start `adfjobs` (or within GUI: click SCM logo > jobs)
  - tools > prepare



template file(s)

overwrite template options

geometry files

run directory

Series of ADF Calculations – Scripted Case

- ADF-GUI not always available (e.g. on supercomputers)
- For such cases: ADFprep & ADFreport

```
~$ adfprep -t Template.adf -m Geometry.xyz [options]
```

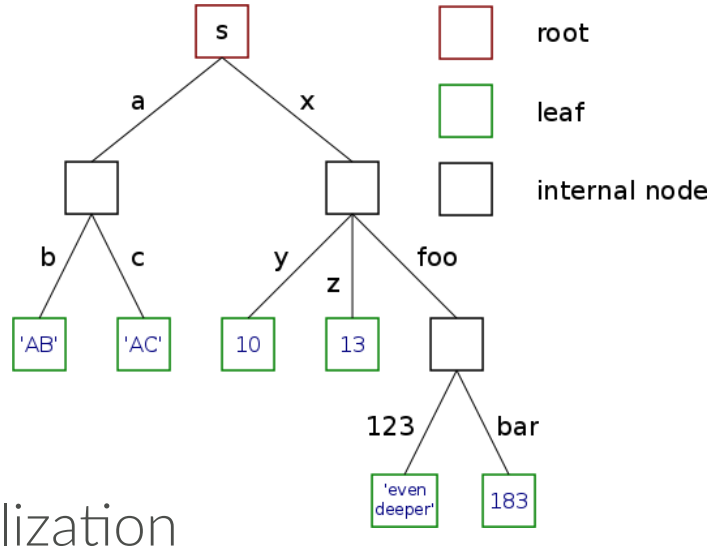
```
~$ adfreport ResultsFile.t21 <Property>
```

- To be used in shell scripts

```
for f in $ADFHOME/examples/adf/BakersetSP/Bakerset/*.xyz
do
  "$ADFBIN/adfprep" -t SP -m "$f" -b DZ -j `basename $f .xyz` runset
done
chmod +x runset

./runset

ls -t -1 *.t21 | while read f
do
    "$ADFBIN/adfreport" "$f" BondingEnergy
done
```

ADF Modelling Suite – Scripting Environments

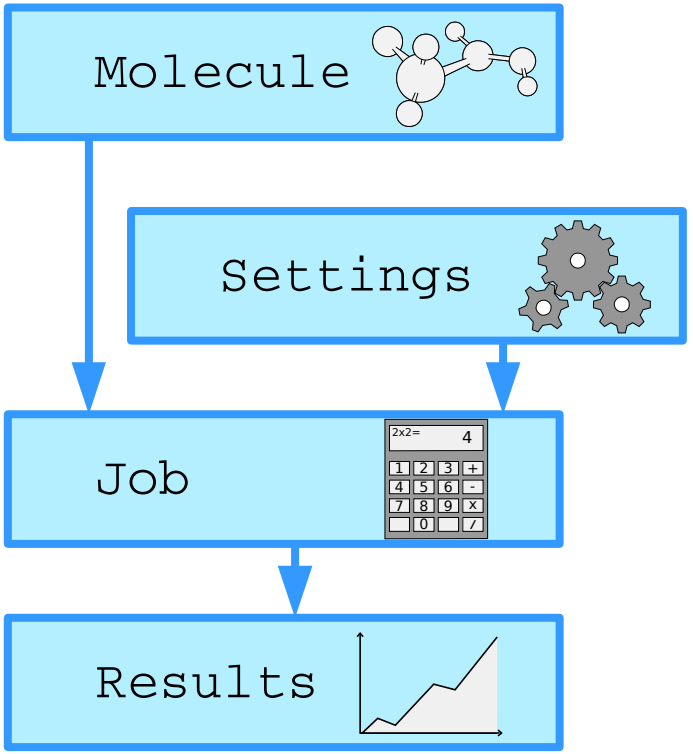
- ADFprep, ADFreport: basic job preparation and results extraction
- ASE – Atomic Simulation Environment (Denmark Technical University)
  - manipulation of atoms
  - geometry optimizers, MD drivers, TS searches …
  - FlexMD, ASE-based scripting for multi-scale MD
- Python
  Library for
  Automating
  Molecular
  Simulations
  - prepare, execute & process jobs
  - manage workflows & workflow parallelization



PLAMS – Scripting in Python

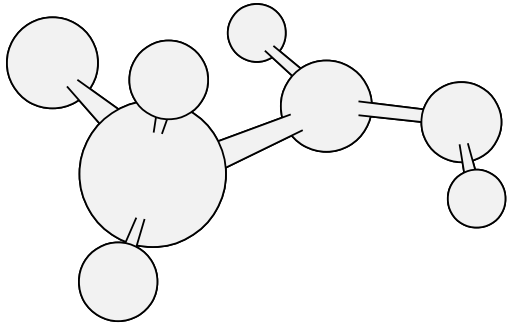
- Shell scripts not very flexible
- Workflow may depend on results
- PLAMS for more complex things!

  Main concept: objects representing
  - Molecule
  - Settings
  - Job
  - Results



SCM

First Steps with PLAMS – Molecule



- Description of molecular system and methods to inspect and change it
- Initialize Molecule object `myMol` from file

```
myMol = Molecule('WaterMolecule.xyz')
```

- Build from individual Atom-objects

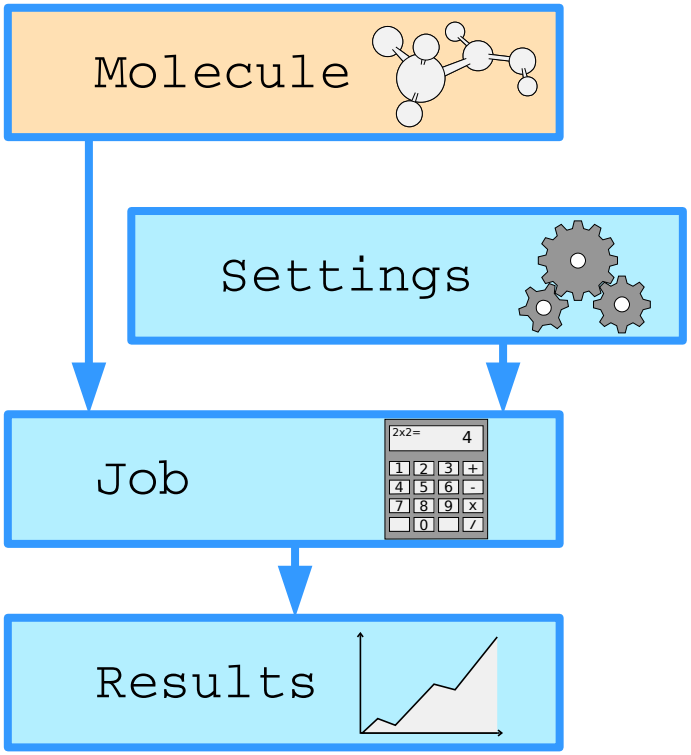
```
myMol = Molecule()
myMol.add_atom(myAtom1)
myMol.add_atom(myAtom2)
myMol.add_atom(myAtom3)
```

yet empty

Atom-objects

whereas:

```
myAtom1 = Atom(symbol='O', coords=(0,0,0))
myAtom2 = ...
```



SCM

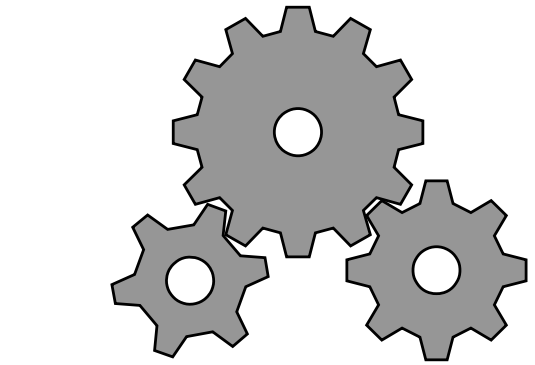
First Steps with PLAMS – Settings

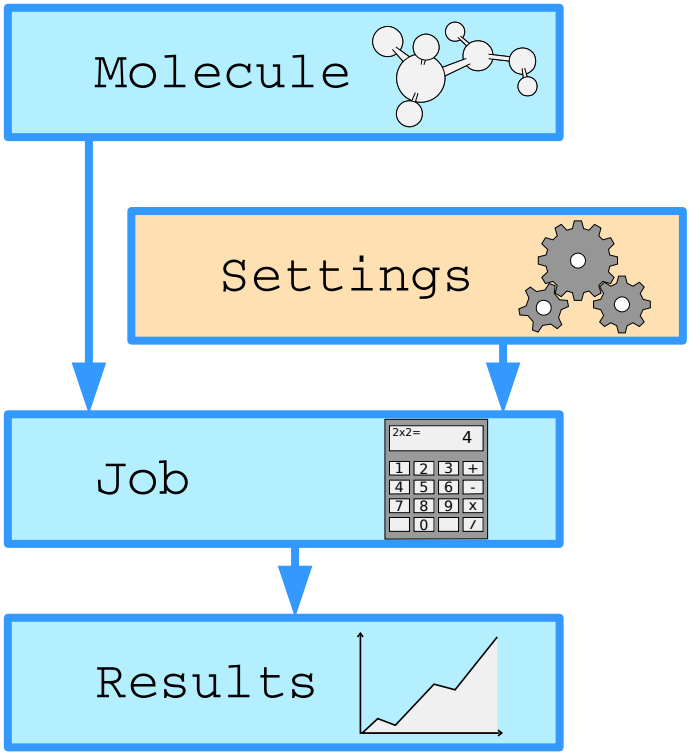
- Initialize Settings object `mySet`
  add options after that

```
mySet = Settings()
mySet.input.basis.type          = 'DZ'
mySet.input.NumericalQuality    = 'Basic'
mySet.input.XC.GGA              = 'PBE'
mySet.input.geometry
```

- Translates into ADF input blocks:

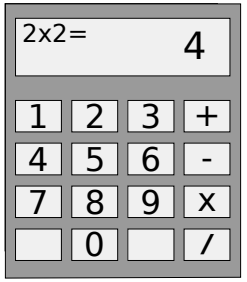
```
Basis
  Core None
  Type DZ
End

Geometry
End

Numericalquality Basic

Xc
  Gga PBE
End
```





SCM

First Steps with PLAMS – Job



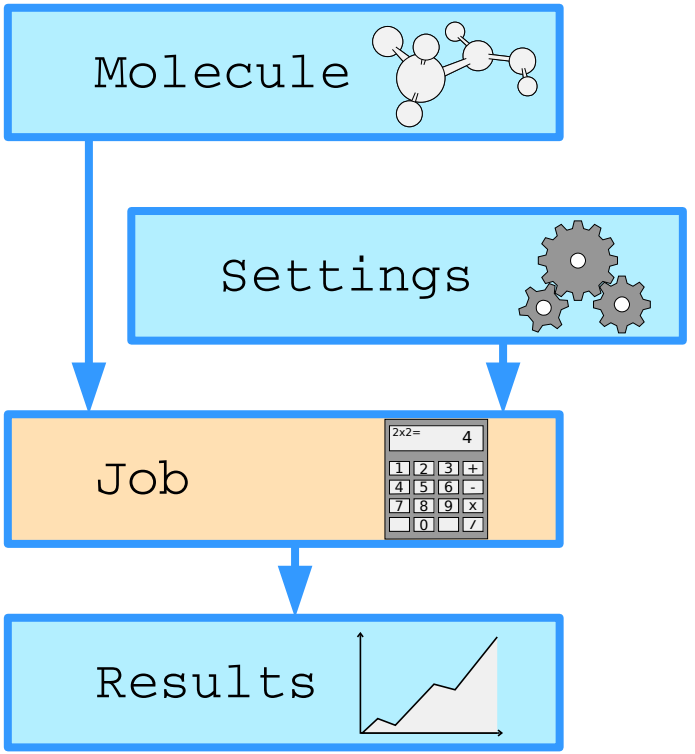
- Initialize Job object `myJob`

```
myJob = ADFJob(molecule=myMol, settings=mySet)
```

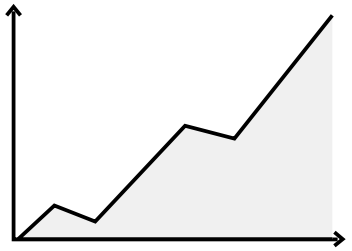
  - `MyJob` adsorbs `myMol` and `mySet`
  - Also available: `BANDJob`, `DFTBJob`, etc.

- Start computation by invoking

```
myJob.run()
```



First Steps with PLAMS – Results



- `myJob.run()` also creates Results object

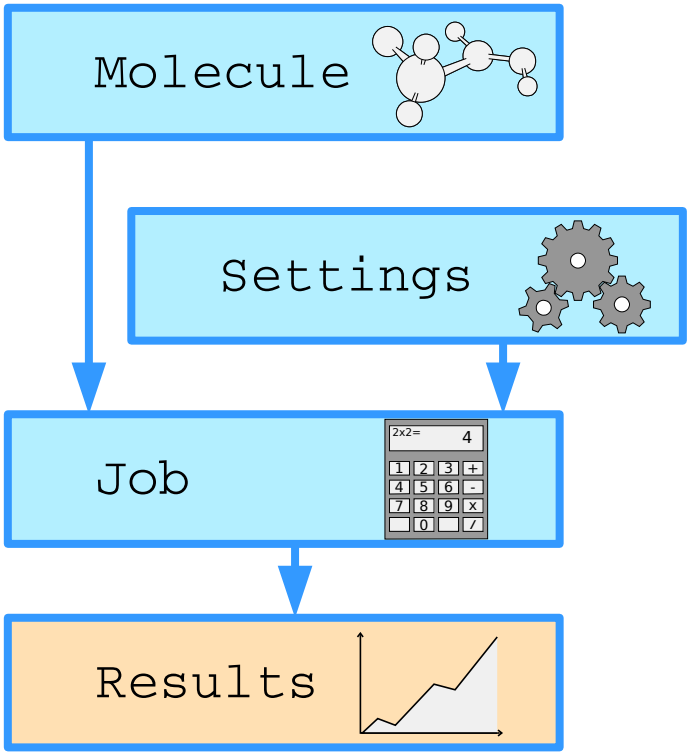
```
MyRes = myJob.run()
```

- Different methods to obtain results

```
energy = myRes.readkf('Energy','Bond Energy')
```

- Alternatively

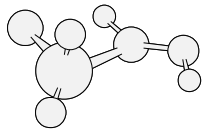
```
myJob.run()
myRes = myJob.results
```

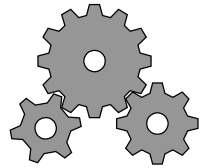


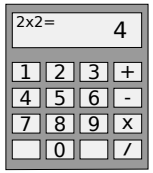
SCM

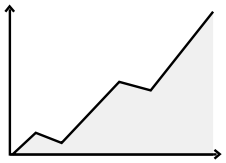
First Steps with PLAMS – Demo Script

```
myMol = Molecule('WaterMolecule.xyz')

mySet = Settings()
mySet.input.basis.type         = 'DZ'
mySet.input.NumericalQuality   = 'Basic'
mySet.input.XC.GGA             = 'PBE'
mySet.input.geometry

myJob = ADFJob(molecule = myMol, settings = mySet)
myJob.run()

energy = myJob.results.readkf('Energy', 'Bond Energy')
optgeo = myJob.results.get_molecule('Geometry', 'xyz')
optang = optgeo.atoms[0].angle(optgeo.atoms[1],optgeo.atoms[2])
print('Energy:     ' +str(energy)+' au')
print('HOH angle:  '+str(optang)+' rad')
```









First Steps with PLAMS – Getting Started

- From PLAMS master script

```
~$ $ADFBIN/plams MyScript1.py
```

  - standalone variant
  - uses correct python version, paths, initialization, and finalization

- As library

```
~$ $ADFBIN/startpython MyScript2.py
```

  - `MyScript2.py`:

```
from scm.plams import *

init()

#
# your actual script goes here!
#

finish()
```

SCM

Documentation, Manuals, other Resources

- Workshop website: hands-on material
  scm.com/collaborations/eu-projects/tccm-adf-tutorial
- SCM scripting documentation
  scm.com/documentation/Scripting
- Atomic Simulation Environment
  wiki.fysik.dtu.dk/ase
- Python language
  python.org/doc

SCM