Electronic structure calculations with the GPAW code

- Pseudo-potentials
- Projector Augmented Wave method
  - The theory
  - Approximations
- GPAW
  - Running calculations with ASE and GPAW
  - Finite-difference, LCAO and plane waves
  - Iterative diagonalization
  - Scaling with system size
Chemical properties are determined largely by valence electrons.

Strong Coulomb potential of nuclei leads to sharply varying wave functions.

Replace the real potential with a pseudo-potential reproducing the correct wave functions outside chosen radius.
Appelbaum-Hamann\textsuperscript{1} potential ($v_1 = 3.042$, $v_2 = -1.372$, $\alpha = 0.6102$):

$$\tilde{v}_{AH}(r) = (v_1 + v_2 r^2) e^{-\alpha r^2} - 4\text{erf}(\sqrt{\alpha} r)/r$$

\textsuperscript{1}Joel A. Appelbaum and D. R. Hamann, PRB 8, 1777 (1973)
Non-local PP’s

- Kleinman-Bylander\(^2\) separable form:

\[
\hat{V} = \hat{V}_{\text{local}} + \sum_{\ell m} \langle \delta v_{\ell m} | \delta v_{\ell m} \rangle
\]

- Ultra-soft PP’s\(^3\): Pseudo wave functions need not be normalized - this allows for ultra-soft wave functions

---


Projector Augmented Wave method

- Access to full all-electron wave-functions, density and potential
- An exact all-electron formalism
- *Pseudo-potentials done right*: PAW contains USPP and NCPP theory

The PAW method was invented by Peter Blöchl in 1994:

The PAW transformation maps nice and smooth wave functions to all-electron wave functions with cusps and nasty oscillatory behavior near the nucleus:

\[ \psi_n(\vec{r}) = \hat{\tau} \tilde{\psi}_n(\vec{r}) \]

\[ \hat{\tau} = \hat{1} + \sum_a \sum_i (|\phi_i^a\rangle - |\tilde{\phi}_i^a\rangle) \langle \tilde{p}_i^a |. \]

- Projector functions: \( \tilde{p}_{n\ell m}^a(\vec{r}) = 0 \) for \( r > r_c^a \).
- All-electron partial waves: \( \phi_{n\ell m}^a(\vec{r}) \).
- Pseudo partial waves: \( \tilde{\phi}_{n\ell m}^a(\vec{r}) = \phi_{n\ell m}^a(\vec{r}) \) for \( r > r_c^a \).
- \( \langle \tilde{\phi}_{n\ell m}^a | \tilde{p}_{n'\ell' m'}^a \rangle = \delta_{nn'} \delta_{\ell\ell'} \delta_{mm'} \)
- \( \sum_{n\ell m} |\tilde{\phi}_{n\ell m}^a\rangle \langle \tilde{p}_{n\ell m}^a | = \hat{1} \) for \( r < r_c^a \) (completeness relation)

Fixed points for the transformation:

\[ \hat{\tau} \tilde{\phi}_i^a = \phi_i^a \]
Note
The pseudo 5d is not normalized
Platinum 6s orbital

\[ \phi_{6s} = \tilde{\phi}_{6s} + \phi^a_{6s} - \tilde{\phi}^a_{6s} \]
\[ |\psi_n\rangle = |\tilde{\psi}_n\rangle + \sum_a \sum_i (|\phi_i^a\rangle - |\tilde{\phi}_i^a\rangle) \langle \tilde{p}_i^a | \tilde{\psi}_n \rangle \]

Take any local or semi-local operator \( \hat{O} \) (the kinetic energy, a local potential, the electron density ...):

\[
\langle \psi_n | \hat{O} | \psi_n \rangle = \langle \tilde{\psi}_n | \hat{O} | \tilde{\psi}_n \rangle + \sum_{abij} \langle \tilde{\psi}_n | \tilde{p}_i^a \rangle (\langle \phi_i^a | - \langle \tilde{\phi}_i^a | \rangle) \hat{O} (| \phi_j^b \rangle - | \tilde{\phi}_j^b \rangle) \langle \tilde{p}_j^b | \tilde{\psi}_n \rangle + \sum_{ai} (\langle \tilde{\psi}_n | \hat{O} (| \phi_i^a \rangle - | \tilde{\phi}_i^a \rangle) \langle \tilde{p}_i^a | \tilde{\psi}_n \rangle + \text{c.c.})
\]

Use that \( \langle \phi_i^a | - \langle \tilde{\phi}_i^a | = 0 \) outside the augmentation sphere of atom \( a \) and use this expansion inside the spheres: \( |\tilde{\psi}_n\rangle = \sum_i |\tilde{\phi}_i^a\rangle \langle \tilde{p}_i^a | \tilde{\psi}_n \rangle \):

\[
\langle \psi_n | \hat{O} | \psi_n \rangle = \langle \tilde{\psi}_n | \hat{O} | \tilde{\psi}_n \rangle + \sum_{aij} \langle \tilde{\psi}_n | \tilde{p}_i^a \rangle (\langle \phi_i^a | \hat{O} | \phi_j^a \rangle - \langle \tilde{\phi}_i^a | \hat{O} | \tilde{\phi}_j^a \rangle) \langle \tilde{p}_j^a | \tilde{\psi}_n \rangle.
\]
From last slide:

\[ \langle \psi_n | \hat{O} | \psi_n \rangle = \langle \tilde{\psi}_n | \hat{O} | \tilde{\psi}_n \rangle + \sum_{aij} \langle \tilde{\psi}_n | \tilde{p}_a^i \rangle (\langle \phi_a^i | \hat{O} | \phi_a^j \rangle - \langle \tilde{\phi}_a^i | \hat{O} | \tilde{\phi}_a^j \rangle) \langle \tilde{p}_a^j | \tilde{\psi}_n \rangle. \]

Often, we are interested in \( \sum_n f_n \langle \psi_n | \hat{O} | \psi_n \rangle \). We introduce the atomic density matrix:

\[ D_{ij}^a = \sum_n \langle \tilde{\psi}_n | \tilde{p}_a^i \rangle f_n \langle \tilde{p}_a^j | \tilde{\psi}_n \rangle \]

We now have:

\[ \sum_n f_n \langle \psi_n | \hat{O} | \psi_n \rangle = \sum_n f_n \langle \tilde{\psi}_n | \hat{O} | \tilde{\psi}_n \rangle + \sum_{aij} D_{ij}^a (\langle \phi_a^i | \hat{O} | \phi_a^j \rangle - \langle \tilde{\phi}_a^i | \hat{O} | \tilde{\phi}_a^j \rangle). \]
Kinetic energy

\[ E_{\text{kin}} = \tilde{E}_{\text{kin}} + \sum_a (E_{\text{kin}}^a - \tilde{E}_{\text{kin}}^a), \]

where

\[ \tilde{E}_{\text{kin}} = -\frac{1}{2} \sum_n f_n \int d\tilde{r} \tilde{\psi}_n^* \nabla^2 \tilde{\psi}_n \]

\[ E_{\text{kin}}^a = -\frac{1}{2} \sum_{ij} D_{ij}^a \int d\tilde{r} \phi_i^a \nabla^2 \phi_j^a - \frac{1}{2} \sum_c \text{core} \int d\tilde{r} \phi_c^a \nabla^2 \phi_c^a \]

\[ \tilde{E}_{\text{kin}}^a = -\frac{1}{2} \sum_{ij} D_{ij}^a \int d\tilde{r} \tilde{\phi}_i^a \nabla^2 \tilde{\phi}_j^a \]
With frozen core density \( n_c^a(a) \) and pseudo core density \( \tilde{n}_c^a(r) = n_c^a(r) \) for \( r < r_c^a \) we have:

\[
    n^a = \sum_{ij} D^a_{ij} \phi_i^a \phi_j^a + n_c^a,
\]

\[
    \tilde{n}^a = \sum_{ij} D^a_{ij} \tilde{\phi}_i^a \tilde{\phi}_j^a + \tilde{n}_c^a,
\]

\[
    \tilde{n} = \sum_n f_n |\tilde{\psi}_n|^2 + \sum_a \tilde{n}_c^a,
\]

Finally, we get a very simple expression for the all-electron density:

\[
    n = \tilde{n} + \sum_a (n^a - \tilde{n}^a)
\]
Exchange-correlation energy

For semi-local functionals:

\[ E_{xc} = \tilde{E}_{xc} + \sum_a (E_{xc}^a - \tilde{E}_{xc}^a), \]

where

\[ \tilde{E}_{xc} = \int d\tilde{r}\tilde{n}\varepsilon_{xc}[\tilde{n}] \]

\[ E_{xc}^a = \int_{r<r_c^a} d\tilde{r}n^a\varepsilon_{xc}[n^a] \]

\[ \tilde{E}_{xc}^a = \int_{r<r_c^a} d\tilde{r}\tilde{n}^a\varepsilon_{xc}[\tilde{n}^a] \]

Note

The densities \( n^a \) and \( \tilde{n}^a \) are not necessarily spherically symmetric!

Note

No non-linear core correction needed - PAW does the right thing!
Let $Z^a(\vec{r})$ be the nuclear charge for atom $a$. The Coulomb energy is:

$$E_C = \frac{1}{2} \int d\vec{r} d\vec{r}' \frac{(n(\vec{r}) + \sum_a Z^a(\vec{r} - \vec{R}^a)) (n(\vec{r}') + \sum_a Z^a(\vec{r}' - \vec{R}^a))}{|\vec{r} - \vec{r}'|}$$

$$= (n + \sum_a Z^a)^2$$

$$= (\tilde{n} + \sum_a [n^a - \tilde{n}^a + Z^a])^2$$

We add and subtract compensation charges localized inside the augmentation spheres:

$$E_C = (\tilde{n} + \sum_a \tilde{Z}^a + \sum_a [n^a - \tilde{n}^a + Z^a - \tilde{Z}^a])^2$$
The compensation charges are constructed like this:

\[
\tilde{Z}^a(\mathbf{r}) = \sum_{\ell m} Q^a_{\ell m} \tilde{g}^a_{\ell m}(\mathbf{r}),
\]

where \( \tilde{g}^a_{\ell m}(\mathbf{r}) = 0 \) for \( r > r_c^a \):

\[
\tilde{g}^a_{\ell m}(\mathbf{r}) = C_\ell r^\ell \exp(-\alpha^a r^2) Y_{\ell m}(\hat{r}),
\]

The \( Q^a_{\ell m} \)'s are chosen such that \( n^a - \tilde{n}^a + Z^a - \tilde{Z}^a \) has no multipole moments:

\[
\int d\mathbf{r} r^\ell Y_{\ell m}(\hat{r})(n^a - \tilde{n}^a + Z^a - \tilde{Z}^a) = 0
\]
Using $\tilde{\rho} = \tilde{n} + \sum_a \tilde{Z}^a$, $\tilde{\rho}^a = \tilde{n}^a + \tilde{Z}^a$ and $\rho^a = n^a + Z^a$, we get:

$$E_C = (\tilde{n} + \sum_a \tilde{Z}^a + \sum_a [n^a - \tilde{n}^a + Z^a - \tilde{Z}^a])^2$$

$$= (\tilde{\rho} + \sum_a [\rho^a - \tilde{\rho}^a])^2$$

$$= \tilde{\rho}^2 + 2\tilde{\rho} \sum_a (\rho^a - \tilde{\rho}^a) + \sum_{ab} (\rho^a - \tilde{\rho}^a)(\rho^b - \tilde{\rho}^b)$$

Since $\rho^a - \tilde{\rho}^a$ has no multipole moments, we get:

$$E_C = \tilde{\rho}^2 + 2 \sum_a \tilde{\rho}^a (\rho^a - \tilde{\rho}^a) + \sum_a (\rho^a - \tilde{\rho}^a)^2$$

$$= \tilde{\rho}^2 + \sum_a (\rho^a)^2 - \sum_a (\tilde{\rho}^a)^2$$
Electrostatic energy

\[ E_C = \tilde{\rho}^2 + \sum_a (\rho^a)^2 - \sum_a (\tilde{\rho}^a)^2 = \tilde{E}_C + \sum_a (E_C^a - \tilde{E}_C^a) \]

\[ \tilde{E}_C = \frac{1}{2} \int d\vec{r} d\vec{r}' \frac{\tilde{\rho}(\vec{r})\tilde{\rho}(\vec{r}')}{|\vec{r} - \vec{r}'|} \]

1) Solve Poisson equation:

\[ \nabla^2 \tilde{\nu}_H = -4\pi \tilde{\rho} \]

\[ \tilde{E}_C = \frac{1}{2} \int d\vec{r} \tilde{\nu}_H \tilde{\rho} \]

2) Fourier transform:

\[ \tilde{E}_C = \frac{1}{2} \sum_G |\tilde{\rho}(G)|^2 / G^2 \]
Total energy:

\[ E = \tilde{E}_{\text{kin}} + \tilde{E}_C + \tilde{E}_{\text{xc}} + \sum_a \Delta E^a(D^a_{ij}) = \tilde{E}[\{\tilde{\psi}_n\}] + \sum_a \Delta E^a(D^a_{ij}) \]

Hamiltonian (\( \delta E / \delta \tilde{\psi}^*_n = f_n \hat{H} \tilde{\psi}_n \)):

\[ \hat{H} = -\frac{1}{2} \nabla^2 + \tilde{v} + \sum_a \sum_{ij} |\tilde{p}^a_i\rangle \Delta H^a_{ij} \langle \tilde{p}^a_j| \]

where \( \tilde{v} = \delta \tilde{E} / \delta \tilde{n} = \tilde{v}^H + \tilde{v}_{\text{xc}} \) and

\[ \Delta H^a_{ij} = \frac{\Delta E^a}{\partial D^a_{ij}} + \sum_{\ell m} \frac{\partial Q^a_{\ell m}}{\partial D^a_{ij}} \int d\vec{r} \tilde{v}^H \tilde{g}^a_{\ell m} \]

(the PAW method is a generalized Kleinman-Bylander non-local pseudo-potential that adapts to the current environment)
Orthogonality

Keep the wave functions orthogonal:

\[ \delta_{nm} = \langle \psi_n | \psi_m \rangle = \langle \tilde{\psi}_n | \hat{T} \hat{T} | \tilde{\psi}_m \rangle = \langle \tilde{\psi}_n | \hat{S} | \tilde{\psi}_m \rangle, \]

where

\[ \hat{S} = \hat{T}^T \hat{T} = \hat{1} + \sum_a \sum_{ij} \langle \tilde{p}^a_i | \Delta S^a_{ij} | \tilde{p}^a_j \rangle \]

and

\[ \Delta S^a_{ij} = \int d\vec{r} (\phi^a_i \phi^a_j - \tilde{\phi}^a_i \tilde{\phi}^a_j) \]

We need to solve:

\[ \hat{H} \tilde{\psi}_n = \varepsilon_n \hat{S} \tilde{\psi}_n \]
Summary of approximations

- Frozen core states
- Finite number of projectors, partial waves and pseudo partial waves
- Overlapping augmentation spheres
- Standard DFT stuff:
  - Occupation number smearing
  - $\vec{k}$-point sampling.
  - XC-functional
from ase import Atoms
h2 = Atoms(...)
from ase.calculators.emt import EMT
h2.calc = EMT()
h2.get_potential_energy()
from ase.optimize import BFGS
opt = BFGS(h2)
opt.run(fmax=0.05)
h2.get_forces()
from gpaw import GPAW, PW
h2.calc = GPAW(mode=PW(300), txt='h2.txt')
h2.center(vacuum=2)
h2.cell
h2.get_forces()
from ase.visualize import view
view(h2)
Alternative to `Atoms(...)`:

```python
from ase.structure import molecule
h2 = molecule('H2')
```

or:

```python
from ase.io import read
h2 = read('H2.xyz')
```

- There is also a `bulk()` function and, and, and ...
- Use ASE’s GUI called `ase-gui`. It understands the same file-formats as ASE’s `read()` function: GPAW’s text output, `.gpw` files, ASE’s `.traj` files and many more.
What is GPAW?

- An implementation of the PAW method
- Wave functions are described using
  1) real-space uniform grids (fd)
  2) atom-centered numerical basis functions (lcao)
  3) plane-waves (pw)
- It’s written in a combination of the Python and C languages
- It’s based on ASE and NumPy (a Python library for manipulating N-dimensional arrays of numbers)
- It uses these libraries for the hard work:
  - BLAS (GPAW does a lot of matrix-matrix multiplications)
  - LAPACK and optionally ScaLAPACK and BLACS
  - MPI (if you want to run GPAW in parallel)
  - FFTW (if you want to do fast plane-wave based calculations)
  - LIBXC (for LDA’s, GGA’s and MGGA’s)
- The license is GPLv3+
Describe wave functions, electron density and the effective potential in real-space on uniform grids:

\[ \vec{r}_{i_1 i_2 i_3} = \left( i_1 / N_1 \right) \vec{a}_1 + \left( i_2 / N_2 \right) \vec{a}_2 + \left( i_3 / N_3 \right) \vec{a}_3, \]

where \( \vec{a}_1, \vec{a}_2 \) and \( \vec{a}_3 \) are the unit cell vectors and we use \( N_1 \times N_2 \times N_3 \) grid points \( (i_\alpha = 0, 1, ..., N_\alpha - 1) \). Use finite-difference approximation for \( \nabla^2 \) and iterative multi-grid based solvers for the Poisson and Kohn-Sham equations.

\[ \frac{\partial^2 f(x)}{\partial x^2} = \sum_{n=-N}^{N} C_n^N f(x + nh) + O(h^{2N+2}) \]
Finite difference

Advantages:
- It’s simple! Only one parameter (grid-spacing).
- No FFT’s. Easy to parallelize using domain decomposition - only neighbor-neighbor communication.

Disadvantages:
- You need many grid-points.
Gold clusters: \( \text{Au}_{561} \) and \( \text{Au}_{102}\text{S}_{44}\text{O}_{88}\text{C}_{308}\text{H}_{220} \). Important to parallelize \((N_e \times N_e)\) matrix operations (ScaLAPACK) and parallelize over states.

**Note**

Don’t use this many cores for the exercises!
LCAO mode

Expand wave function in numerical atom-centered atomic-like orbitals with finite support:

\[ \tilde{\psi} = \sum_a \sum_{n\ell m} c_{n\ell m}^a \Phi_{n\ell m}^a(\vec{r} - \vec{R}^a). \]

- The atomic orbitals are obtained from a free atom in a confining potential well
- Extra basis functions with the same angular momentum
- Polarization functions
- Same PAW approximation as with real-space grids

Advantages:
- \( \langle \tilde{p}^a_i | \Phi_{n'\ell' m'}^{a'} \rangle, \langle \Phi_{n\ell m}^a | \Phi_{n'\ell' m'}^{a'} \rangle \) and \( \langle \Phi_{n\ell m}^a | \nabla^2 | \Phi_{n'\ell' m'}^{a'} \rangle \) can be calculated exactly
- One can do quick’n’dirty calculations

Disadvantages:
- Hard to reach complete basis-set limit.
Plane wave basis set

\[ \tilde{\psi}(r) = \sum_{G < G_c} c_G e^{iG \cdot r}. \]

\[ E_{\text{cut}} = \frac{1}{2} G_c^2 \]

It’s based on FFT and does the projector wave function overlaps in reciprocal space with ZGEMM.

Advantages:
- Fast for not too large systems
- Fast convergence with respect to number of plane-waves

Disadvantages:
- not so flexible boundary conditions
- delocalized basis functions
Iterative diagonalization algorithm (FD and PW mode)

\[ \hat{H}\tilde{\psi}_n = \varepsilon_n\hat{S}\tilde{\psi}_n \]

1) Initial guess for wave functions from LCAO (\(\tilde{\psi}_n\))
2) Orthogonalize wave functions (make sure \(\langle \tilde{\psi}_n | \hat{S} | \tilde{\psi}_m \rangle = \delta_{nm}\))
3) Calculate density, mix\(^4\) and interpolate to finer grid (\(\tilde{n}, D_{ij}^a\))
4) Calculate potential (\(\tilde{\nu}, \Delta H_{ij}^a\))
5) Apply Hamiltonian (\(\hat{H}\tilde{\psi}_n\))
6) Subspace diagonalization (rotate \(\tilde{\psi}_n\) so that \(\langle \tilde{\psi}_n | \hat{H} | \tilde{\psi}_m \rangle = \delta_{nm}\varepsilon_n\))
7) Calculate residuals (\(R_n = \hat{H}\tilde{\psi}_n - \varepsilon_n\hat{S}\tilde{\psi}_n\))
8) Improve wave functions\(^4\) (\(\tilde{\psi}_n + \lambda \hat{P} R_n \rightarrow \tilde{\psi}_n\))
9) Back to (2)

---

Scaling with system size $N$: FD-mode and PW-mode

- Number of atoms: $N_a \sim 100$
- Number of electrons: $N_e \sim 500$
- Number of grid points or plane waves: $N_g \sim 100,000$

Linearly scaling parts of the code:
- Coulomb energy:
  - FD: (Poisson equation) $N_g$
  - PW: (direct method) $N_g \log N_g$
- Exchange-correlation energy and potential: $N_g$
- Density mixing: $N_g$
Scaling with system size $N$ (continued)

**Quadratic:**
- Evaluation of projections $\langle \tilde{p}_i^a | \tilde{\psi}_n \rangle$:
  - FD: $N_a N_e$
  - PW: $N_a N_e N_g$
- Calculation of $-\frac{1}{2} \nabla^2 \tilde{\psi}_n$ and $\tilde{v} \tilde{\psi}_n$:
  - FD: $N_e N_g$
  - PW: $N_e N_g \log N_g$
- Pseudo density $\tilde{n} = \sum_n f_n |\tilde{\psi}_n|^a + \sum_a \tilde{n}_c^a$: $N_e N_g$

**Cubic:**
- Orthogonalization of wave functions: $N_e^2 N_g$ ($S_{nn'}$)
- Subspace diagonalization: $N_e^2 N_g$ ($H_{nn'}$)
- $(N_e \times N_e)$-matrix operations: $N_e^3$
We have generated datasets \((\phi^a_i, \tilde{\phi}^a_i, \tilde{p}^a_i, n^a_c, \tilde{n}^a_c, \bar{v}^a)\) for most elements, but:

**Important**

it’s up to you to test them!

- Do test calculations and compare to reference numbers

```
$ gpaw-setup Si -f BLYP  # dataset for BLYP-Si
$ gpaw-setup Si -r 2.0   # change cutoff
$ gpaw-setup -h          # HELP!
```
Exercises

- Surface energy, diffusion, band structure, Wannier functions, NEB, magnetism, DOS, STM, databases, vibrations, stress tensor, transport, TDDFT.
- EELS, GW and RPA correlation: For these exercises you need the density response function: $\chi_{GG'}^0(q, \omega)$
  1. Do a ground-state calculation
  2. Fixed potential and set up Hamiltonian and overlap matrices
  3. Find all eigenvectors using (Sca)Lapack

```python
from gpaw import GPAW
calc = GPAW('abc.gpw', txt=None)
calc.diagonalize_full_hamiltonian(nbands=...)
calc.write('abc.all.gpw', 'all')
```
Where is what?

$\text{GPAW\_SETUP\_PATH}$:

*PAW-datasets* (*H.LDA.gz*, *He.LDA.gz*, ...) also contains *PBE*, *revPBE*, *RPBE*, *GLLBSC* and *LCAO* basis functions.

$\text{PYTHONPATH}$:

- *The ASE code*.
- *The GPAW code*.
- *\_gpaw.so*: GPAW’s C-extensions for Python.

$\text{PATH}$:

- *gpaw-python*: MPI-enabled Python interpreter with GPAW’s C-code inside.
Finally ...

- Check out:
  - http://wiki.fysik.dtu.dk/ase
  - http://wiki.fysik.dtu.dk/gpaw

- Thanks to the GPAW-team:

- Talk to us on the #gpaw channel on irc.freenode.net or on our mailing lists: gpaw-developers, ase-developers, gpaw-users and ase-users.

- Thank you for your attention

- Questions?
The egg-box effect

Translation of hydrogen in empty space with $h = 0.19 \text{ Å}$:

Fourier filtering\(^5\) of $\tilde{p}_i$ and $\tilde{v}$:

- Fourier transform $\tilde{p}_i(r)/m(r)$, where $m(r)$ is a mask function that goes smoothly to zero at a chosen cutoff radius.
- Cut off high frequency components.
- Transform back and multiply by $m(r)$.

\(^5\)Mask-function real-space implementations of nonlocal pseudo-potentials, Phys. Rev. B 64, 201107(R) Published 6 November 2001, Lin-Wang Wang
PAW atomic datasets in one slide

- Solve \((-\frac{1}{2} \nabla + v - \varepsilon_{nl})\phi_{nl} = 0\)
- Construct \(\tilde{\phi}_{nl}\): smooth continuation of \(\phi_{nl}\) for \(r < r_c\)
- Construct \(\tilde{n}_c\): smooth continuation of core density \(n_c\) for \(r < r_c\)
- \(\tilde{n} = \tilde{n}_c + \sum_{nl} f_{nl} \tilde{\phi}_{nl}^2\)
- \(\tilde{\rho} = \tilde{n} + Qe^{-\alpha r^2}, \int d\mathbf{r} \tilde{\rho} = 0\)
- Construct \(\tilde{v}\) somehow (several possibilities)
- Calculate projector functions: \(\tilde{p}_{nl} \propto (-\frac{1}{2} \nabla + \tilde{v} - \varepsilon_{nl})\tilde{\phi}_{nl}\)
- Unscreen: \(\bar{v} = \tilde{v} - \int d\mathbf{r}' \tilde{\rho}(\mathbf{r}')/|\mathbf{r} - \mathbf{r}'| - v_{xc}[\tilde{n}]\)
- Add more projectors
- Check logarithmic derivatives and pseudo-atom eigenvalues