

Localized atomic orbitals in GPAW

or as we like to call it:
“LCAO mode”

Ask Hjorth Larsen

Nano-bio Spectroscopy Group
Departamento de Física de Materiales
Universidad del País Vasco UPV/EHU

May 21, 2013

Real-space mode, `mode='fd'`

- ▶ Very accurate
- ▶ Excellent parallelization (\mathbf{k} -points, domains, bands)
- ▶ Can be a bit expensive at times

LCAO mode, `mode='lcao'`

- ▶ Very efficient, particularly for large systems
- ▶ Good parallelization (\mathbf{k} -points, domains, bands/orbitals)
- ▶ Not all that accurate

Planewave mode, `mode='pw'`

- ▶ Very accurate, planewaves are very neat
- ▶ But less parallel than 'fd' (\mathbf{k} -points, bands only)

Localized basis sets

- ▶ Expand pseudowavefunctions in fixed orbitals:

$$|\tilde{\psi}_n\rangle = \sum_{\mu} |\Phi_{\mu}\rangle c_{\mu n}$$

- ▶ Now $c_{\mu n}$ are the variational variables, and $|\Phi_{\mu}\rangle$ are fixed localized functions
- ▶ Derive new Kohn–Sham equations and solve:

$$\sum_{\nu} H_{\mu\nu} c_{\nu n} = \sum_{\nu} S_{\mu\nu} c_{\nu n} \epsilon_n$$

Thanks to Marco Vanin for the collaboration back in the days

Advantages

- ▶ More locality → better scaling of many operations
- ▶ Smaller dimension of Hamiltonian → we can use direct solver
- ▶ Direct solver → easier to converge (fewer steps)
- ▶ Small basis allows e.g. Green's function based transport calculations

Disadvantages

- ▶ Basis set is much less “complete” than real-space/planewaves
- ▶ Binding energies not so accurate
- ▶ (Alternatively: Spend ages choosing good basis functions)
- ▶ No simple way to crank up precision

Using LCAO mode

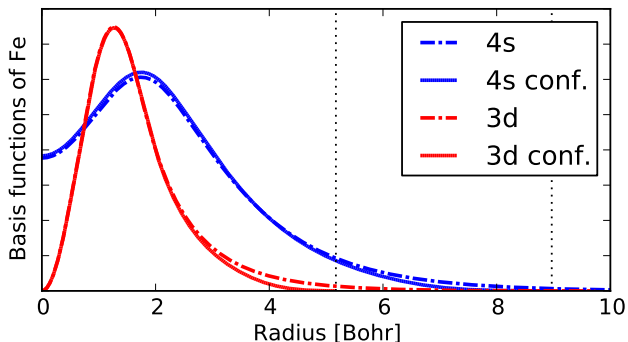
```
from ase.structure import molecule
from gpaw import GPAW

system = molecule('H2O')
system.center(vacuum=5.0)

calc = GPAW(mode='lcao', # important
            basis='dzp', # also important
            h=0.18,      # (the usual stuff)
            xc='PBE')

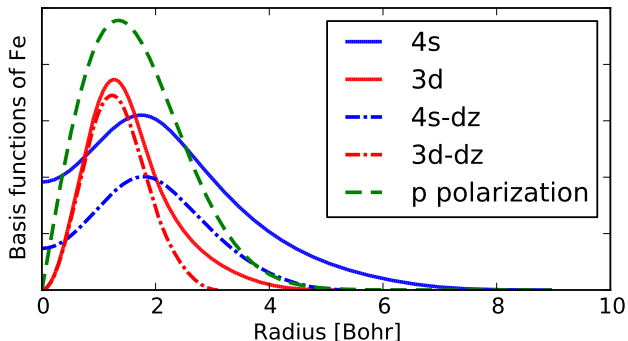
system.set_calculator(calc)
system.get_potential_energy()
```

Choosing basis functions — atomic orbitals



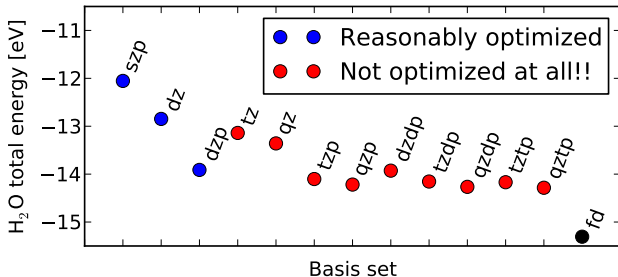
- ▶ Solve spherical Kohn–Sham equations for isolated atom
- ▶ Use external potential to confine wavefunctions within some cutoff
- ▶ Cutoff defined by requiring that each orbital increases in energy a bit (e.g. 0.1 eV)

Choosing basis functions



- ▶ Obtain one atomic orbital from each valence state l, n by solving radial atomic equation
- ▶ Add more functions with different radial parts
- ▶ Add more functions with different angular momentum

Basis set quality



- ▶ Convergence of total energies with basis
- ▶ Basis sets are optimized only up to dzp!
- ▶ LCAO is better suited for structures
- ▶ (Note: Energy *differences* converge faster)

Calculation procedure

Solve $\nabla^2 \tilde{v}_H(\mathbf{r}) = -4\pi \tilde{\rho}(\mathbf{r}), \quad \tilde{\rho}(\mathbf{r}) = \tilde{n}(\mathbf{r}) + \text{atoms...} \quad \mathcal{O}(N)$

Calculate $V_{\mu\nu} = \int \Phi_\mu^*(\mathbf{r}) [\tilde{v}_H(\mathbf{r}) + \tilde{v}_{xc}(\mathbf{r}) + \bar{v}(\mathbf{r})] \Phi_\nu(\mathbf{r}) d\mathbf{r} \quad \mathcal{O}(N)$

Calculate $H_{\mu\nu} = T_{\mu\nu} + V_{\mu\nu} + \sum_{aij} P_{i\mu}^{a*} \Delta H_{ij}^a P_{j\nu}^a \quad \mathcal{O}(N)$

with: $T_{\mu\nu} = \langle \Phi_\mu | \hat{T} | \Phi_\nu \rangle, \quad P_{i\mu}^a = \langle \tilde{p}_i^a | \Phi_\mu \rangle \quad \mathcal{O}(N)$

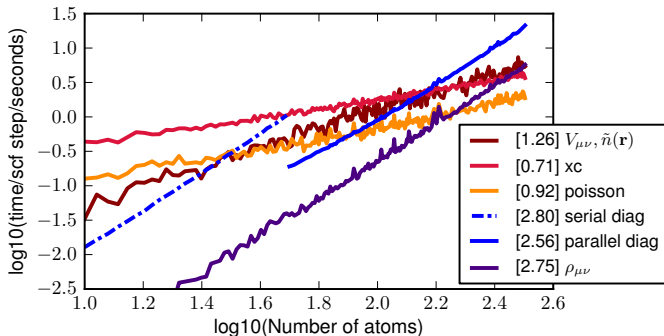
Solve $\sum_\nu H_{\mu\nu} c_{\nu n} = \sum_\nu S_{\mu\nu} c_{\nu n} \epsilon_n \quad \mathcal{O}(N^3)$

Calculate $\rho_{\nu\mu} = \sum_n c_{\nu n} f_n c_{\mu n}^* \quad \mathcal{O}(N^3)$

Calculate $\tilde{n}(\mathbf{r}) = \sum_{\mu\nu} \Phi_\mu^*(\mathbf{r}) \Phi_\nu(\mathbf{r}) \rho_{\nu\mu} + \tilde{n}_{\text{core}}(\mathbf{r}) \quad \mathcal{O}(N)$

(Repeat as necessary)

Operation	Parallelization	Complexity
Multigrid Poisson	\mathbf{r}	$\mathcal{O}(N)$
Density $\tilde{n}(\mathbf{r})$	\mathbf{r}, σ	$\mathcal{O}(N)$
XC $\tilde{v}_{xc}(\mathbf{r})$	\mathbf{r}, σ	$\mathcal{O}(N)$
Potential $V_{\mu\nu}$	$\nu, \mathbf{r}, \sigma, k$	$\mathcal{O}(N)$
Diagonalize $H_{\mu\nu}$	μ, ν, σ, k	$\mathcal{O}(N^3)$
Density matrix $\rho_{\mu\nu}$	μ, ν, σ, k	$\mathcal{O}(N^3)$



Parallelization of $\mathcal{O}(N^3)$ operations

0	2	4	6	0	2	4	6
1	3	5	7	1	3	5	7
0	2	4	6	0	2	4	6
1	3	5	7	1	3	5	7
0	2	4	6	0	2	4	6
1	3	5	7	1	3	5	7

0	2	4	6
---	---	---	---

- ▶ Left: 2D block cyclic layout for diagonalization

$$H_{\mu\nu}, c_{\mu n}, \rho_{\mu\nu}$$

- ▶ Right: 1D columnated layout

$$\sum_{\mu\nu} \Phi_{\mu}^*(\mathbf{r}) \Phi_{\nu}(\mathbf{r}) \rho_{\mu\nu} \rightarrow \tilde{n}(\mathbf{r})$$

- ▶ Must transfer back and forth between two layouts

Parallelization modes

- ▶ Parallelize over \mathbf{k} -points/spins first (N_K cores)
- ▶ Parallelize over domains next ($N_D = N_x \times N_y \times N_z$)
- ▶ Parallelize over bands to use less memory (N_B)
- ▶ Total number of CPUs should be $N_K \times N_D \times N_B$
- ▶ CPUs for ScaLAPACK are taken within groups of $N_D \times N_B$

Rules of thumb for LCAO calculations

- ▶ ScaLAPACK might help beyond 30–60 atoms
- ▶ ScaLAPACK operations require high bandwidth (run within same node/infiniband)
- ▶ Parallelize a lot more over domains than bands
- ▶ Band parallelization helps reduce memory usage
- ▶ Band parallelization is useless without ScaLAPACK

Parallelization

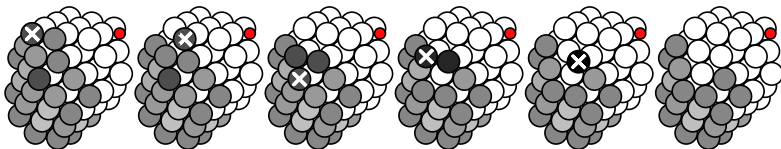
```
from ase.io import read
from gpaw import GPAW

system = read('hundredsofatoms.traj')

# Assume we have 32 cores
calc = GPAW(mode='lcao', basis='dzp',
            parallel=dict(domain=(2, 2, 4),
                          band=2,
                          sl_default=(4, 2, 64)),
            nbands = len(system) * 6,
            xc='PBE')
system.set_calculator(calc)
system.get_potential_energy()
```

Example application

- ▶ Large numbers of similar calculations to investigate trends
- ▶ Study of reactivity of metal clusters of 20–200 atoms



Application: Size effects in transition metal clusters

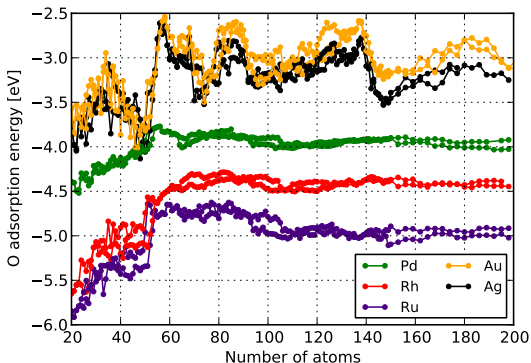


Figure: Adsorption energies of oxygen on different transition metal clusters

s^1d^7	s^1d^8	s^1d^9	s^1d^{10}	s^2d^{10}
Fe	Co	Ni	Cu	Zn
Ru	Rh	Pd	Ag	Cd
Os	Ir	Pt	Au	Hg

- ▶ Smooth variations can be attributed to “geometric effects”
- ▶ Noble metals show strong oscillations from electronic shell structure

Conclusions

- ▶ LCAO calculations are quite fast
- ▶ Do mind the parallelization options
- ▶ We should work on improving basis sets
- ▶ Maybe some day we should implement sparse / iterative eigensolver
- ▶ Thank you for listening!