MAX DRIVING THE EXASCALE TRANSITION

www.flapw.de

fleur

Description

FLEUR is an open-source all-electron density functional theory (DFT) code implementing the full-potential linearized augmented-plane-wave (FLAPW) method. As such, FLEUR does not depend on the pseudopotential approximation and is able to provide materials properties for complex solids composed out of all chemical elements. Special features of the code include the treatment of surfaces in a special film mode, the calculation of magnetic interaction parameters and complex magnetic structures and the inclusion of relativistic effects manifest in the spin-orbit coupling.

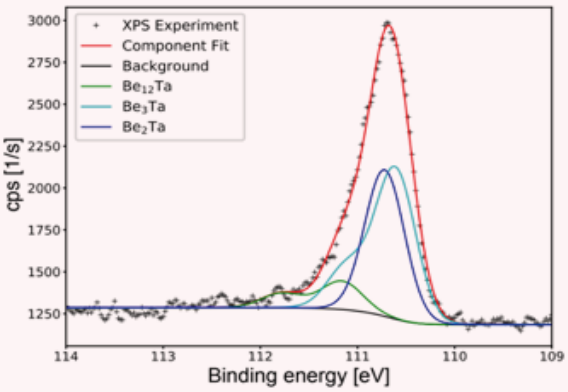
Level of theory

- All-electron DFT
- Standard LDA/GGA exchange correlation treatment, LDA+U, Hybrid functionals
- Fully self-consistent and force-theorem treatment of magnetic properties
- Collinear and noncollinear magnetism
- Efficient description of spin spirals
- Scalar-relativistic description or additional inclusion of spin-orbit coupling

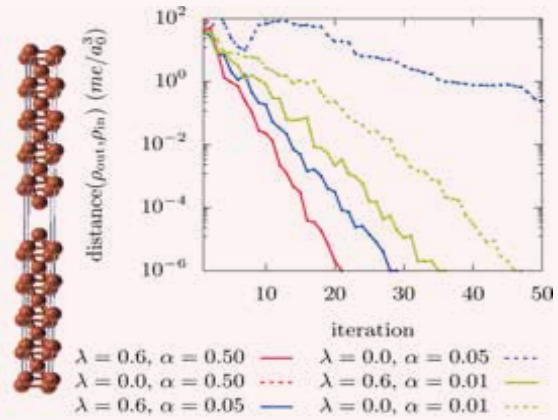
Features

- Electronic properties: bandstructure, DOS; including orbital projections
- Forces, structural relaxations
- Film mode with semi-infinite vacuum simulating surfaces
- Interfaces to AiiDA, libXC, Wannier90
- Band unfolding
- EELS, MCD spectra

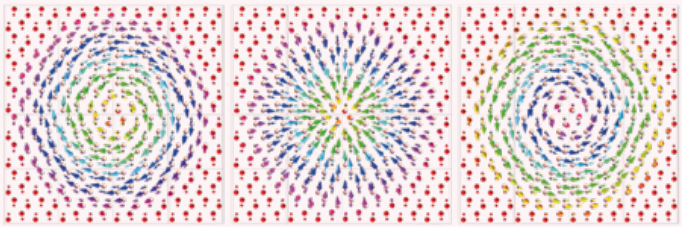
Case studies



High-throughput all-electron density functional theory simulations for a data-driven chemical interpretation of X-ray photoelectron spectra.
[Broeder, PhD thesis, RWTH Aachen (2020)]



Employing the Kerker preconditioner to stabilize and speed-up convergence for SCF calculations on large supercells.
[M. Winkelmann et al., Phys. Rev. B (2020)]



Investigating complex magnetic structures in DFT without relying on model-based approaches.
[U. Alekseeva, work in progress]

MAX DRIVING THE EXASCALE TRANSITION

www.flapw.de

fleur

Parallelization and HPC performance

A main goal of the development of FLEUR is the possibility to efficiently use a large range of computational resources. It can be deployed on personal laptops as well as on the largest supercomputers. This is achieved by optimizing single-node performance using, e.g., multithreading and vectorization and multiple levels of distributed-memory parallelism connected by MPI communication. Interfacing to highly optimized mathematical libraries like LAPACK, ScaLAPACK, ELPA, or FFTW allows to benefit from their performance. Key computational kernels have been ported to GPUs using the OpenACC framework.

Standard DFT-GGA calculation
Outer level, innermost task on single GPU

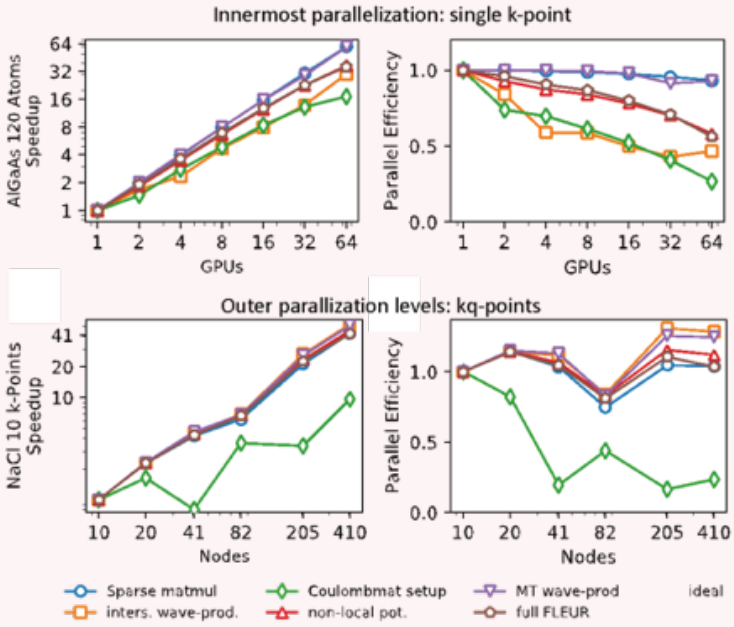
160 k-points —x5→ 800 k-points

Nodes	Time	Scaling
10	434 s	--
20	222 s	1.95
40	121 s	3.59

Nodes	Time	Scaling
50	438 s	--
100	228 s	1.92
200	124 s	3.53

Very good weak scaling, good strong scaling

For standard DFT calculations, the k-point parallelism can be utilized to achieve nearly perfect scaling behavior. In addition, an innermost level distributing the eigenvalue problem on several nodes or GPUs can be used.



Parallel speedup and parallel efficiency of the hybrid functionals implementation in FLEUR. The figures on the top show the innermost parallelization layer while the lower two figures the additional parallelization over k and q points. Both layers can also be combined to utilize several thousand GPUs.

Support

Website

User Forum

Interactive tutorial

Open source code repository with issue tracker

References

- *U. Alekseeva et al., Hybrid Parallelization and Performance Optimization of the FLEUR Code: New Possibilities for All-Electron Density Functional Theory, Euro-Par 2018: Parallel Processing (2018).*
- *M. Redies et al., Fast All-Electron Hybrid Functionals and Their Application to Rare-Earth Iron Garnets, Frontiers in Materials **9**, 851458 (2022).*