

Using FLEUR yourself Future of FLEUR

Daniel Wortmann

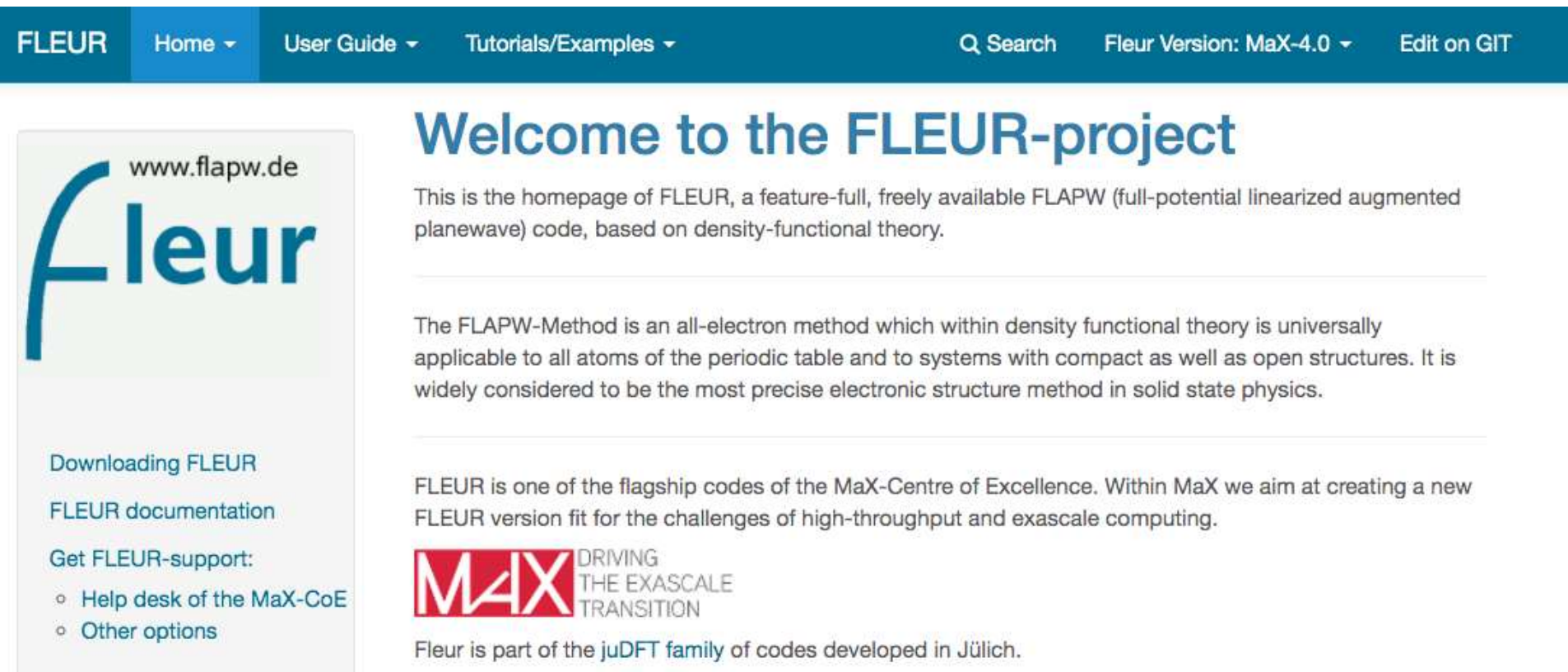
Join the FLEUR community

You are interested in:

- All-electron DFT
- Magnetism, spin-orbit-coupling
- Complex electronic materials
- Surfaces

Main entry point: Homepage

<http://www.flapw.de>



The screenshot shows the homepage of the FLEUR project. At the top is a dark blue navigation bar with the FLEUR logo on the left and links for Home, User Guide, Tutorials/Examples, Search, Fleur Version: MaX-4.0, and Edit on GIT. Below the navigation bar is a light blue header section with the title 'Welcome to the FLEUR-project'. The main content area is white and contains a paragraph about FLEUR being a feature-full, freely available FLAPW code based on density-functional theory. Below this is a paragraph describing the FLAPW-Method as an all-electron method within density functional theory, applicable to all atoms and systems. Further down is a paragraph stating that FLEUR is one of the flagship codes of the MaX-Centre of Excellence, aiming at creating a new FLEUR version for high-throughput and exascale computing. At the bottom left, there is a sidebar with links for downloading FLEUR, FLEUR documentation, and getting FLEUR-support, including a help desk and other options. At the bottom right, there is a logo for the MaX Centre of Excellence, which is part of the juDFT family of codes developed in Jülich.

FLEUR Home ▾ User Guide ▾ Tutorials/Examples ▾ Q Search Fleur Version: MaX-4.0 ▾ Edit on GIT

Welcome to the FLEUR-project

This is the homepage of FLEUR, a feature-full, freely available FLAPW (full-potential linearized augmented planewave) code, based on density-functional theory.

The FLAPW-Method is an all-electron method which within density functional theory is universally applicable to all atoms of the periodic table and to systems with compact as well as open structures. It is widely considered to be the most precise electronic structure method in solid state physics.

FLEUR is one of the flagship codes of the MaX-Centre of Excellence. Within MaX we aim at creating a new FLEUR version fit for the challenges of high-throughput and exascale computing.

MaX DRIVING THE EXASCALE TRANSITION

Fleur is part of the [juDFT family](#) of codes developed in Jülich.

[www.flapw.de](#)
fleur

Downloading FLEUR
FLEUR documentation
Get FLEUR-support:

- Help desk of the MaX-CoE
- Other options

Join the FLEUR community

You are interested in:

- All-electron DFT
- Magnetism, spin-orbit-coupling
- Complex electronic materials
- Surfaces

Main entry point: Homepage

<http://www.flapw.de>

The screenshot shows the homepage of the FLEUR project. The navigation bar at the top includes links for FLEUR, Home, User Guide, and Tutorials/Examples. The User Guide menu is expanded, showing options like Overview, Quick start, Theoretical background, Basic calculations, Expert knowledge and troubleshooting, Reference, and The Fleur-AiiDA interface. The main content area features the title 'the FLEUR-project' and a description of the code as a feature-full, freely available FLAPW (full-potential linearized augmented plane wave) method within density functional theory. It also mentions that FLEUR is one of the flagship codes of the MaX-Centre of Excellence and is part of the juDFT family of codes developed in Jülich.

FLEUR

Home ▾ User Guide ▾ Tutorials/Examples ▾

Q Search Fleur Version: MaX-4.0 ▾ Edit on GIT

the FLEUR-project

www.flapw.de

Fleur

Overview

- Quick start
- Theoretical background
- Basic calculations
- Expert knowledge and troubleshooting
- Reference
- The Fleur-AiiDA interface

Downloading FLEUR

FLEUR documentation

Get FLEUR-support:

- Help desk of the MaX-CoE
- Other options

FLEUR is one of the flagship codes of the MaX-Centre of Excellence. Within MaX we aim at creating a new FLEUR version fit for the challenges of high-throughput and exascale computing.

MAX DRIVING THE EXASCALE TRANSITION

Fleur is part of the juDFT family of codes developed in Jülich.

Suitable architectures

We do not provide binaries!



To use FLEUR you need to compile yourself:

- Decent Fortran-2003 compiler
- Compatible C-compiler
- Libxml2 library (+include files)
- BLAS and LAPACK libraries
- cmake

Good experience
with:

- Intel toolchain
- Gfortran/gcc
- PGI/NVIDIA
compilers

Suitable architectures

FLEUR can take advantage of additional libraries:

MPI:	needed for parallel execution on distributed memory machines
HDF5:	structured IO of large binary data
LibXC:	many exchange-correlation functionals
SpFFT:	Fast FFT-library
Wannier90:	Generate and use Wannierfunctions in FLEUR

For good performance of FLEUR the dense eigensolver must be tuned to your computing architecture:

(Vendor-)LAPACK, Scalapack, ELPA, Magma, Elemental, ...

Installation procedure

Quick and simple:

1. Download configuration script from our webpage
2. Run downloaded configuration script
3. Compile with 'make'

Installation procedure

Alternative:

1. Download source code (preferably by a 'git clone')
2. Configuration script: './configure.sh'
 - Creates build directory
 - Can download some dependencies
 - Runs cmake to determine compilers and options
 - Use ./configure.sh -h and the documentation
3. Run 'make' in build directory

<https://www.flapw.de/MaX-4.0/katacoda2/>

Life demo:

- Input file for inpgen
- Call of inpgen
- Resulting inp.xml
- Run of FLEUR
- Glance on files...

Support Channels

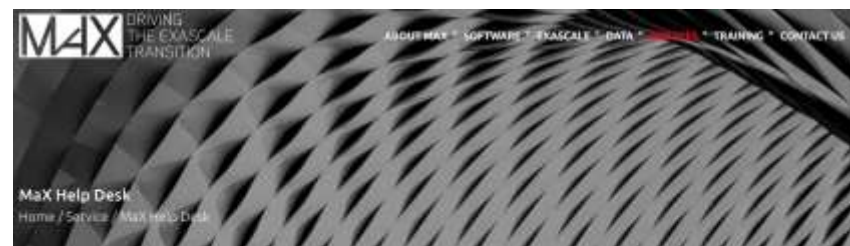
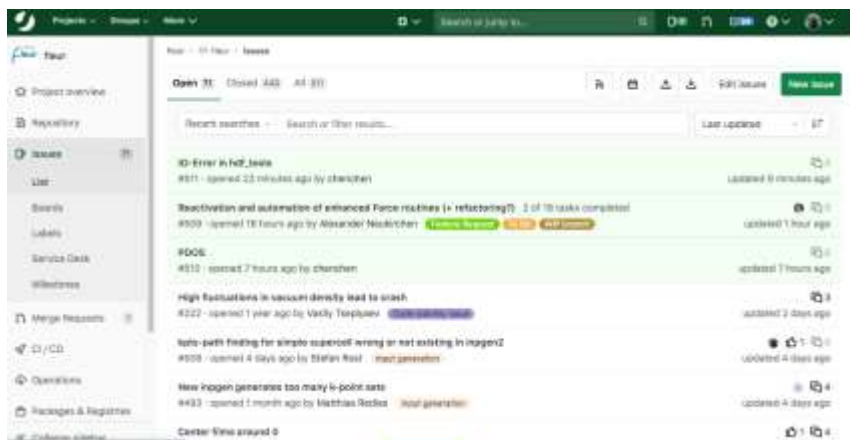
Documentation:
www.flapw.de

**HELP
needed?**

FLEUR Mailing List:
fleur@fz-juelich.de

Issues on
iffgit.fz-juelich.de/fleur/fleur

MaX Helpdesk:
www.max-centre.eu



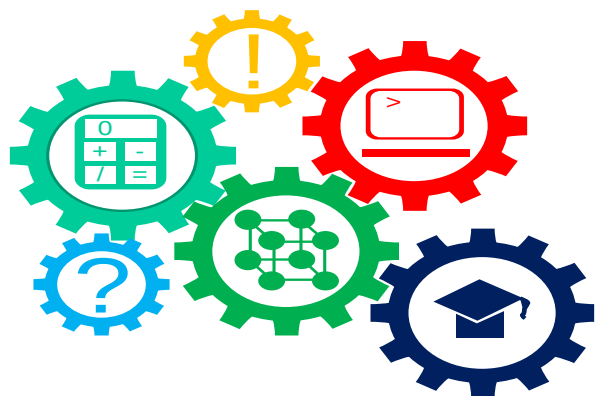
- > Unfortunately still a topic for experienced users
- > CUDA-Fortran and OpenACC needed
=> PGI/NVIDIA compilers

Most problematic:

- Library for dense matrix diagonalization
- Interfaces to ELPA and Magma
- Please contact us directly for support!

The FLEUR AiiDA plugin

- > High throughput computing using FLEUR
- > Full provenance of input, output and calculations



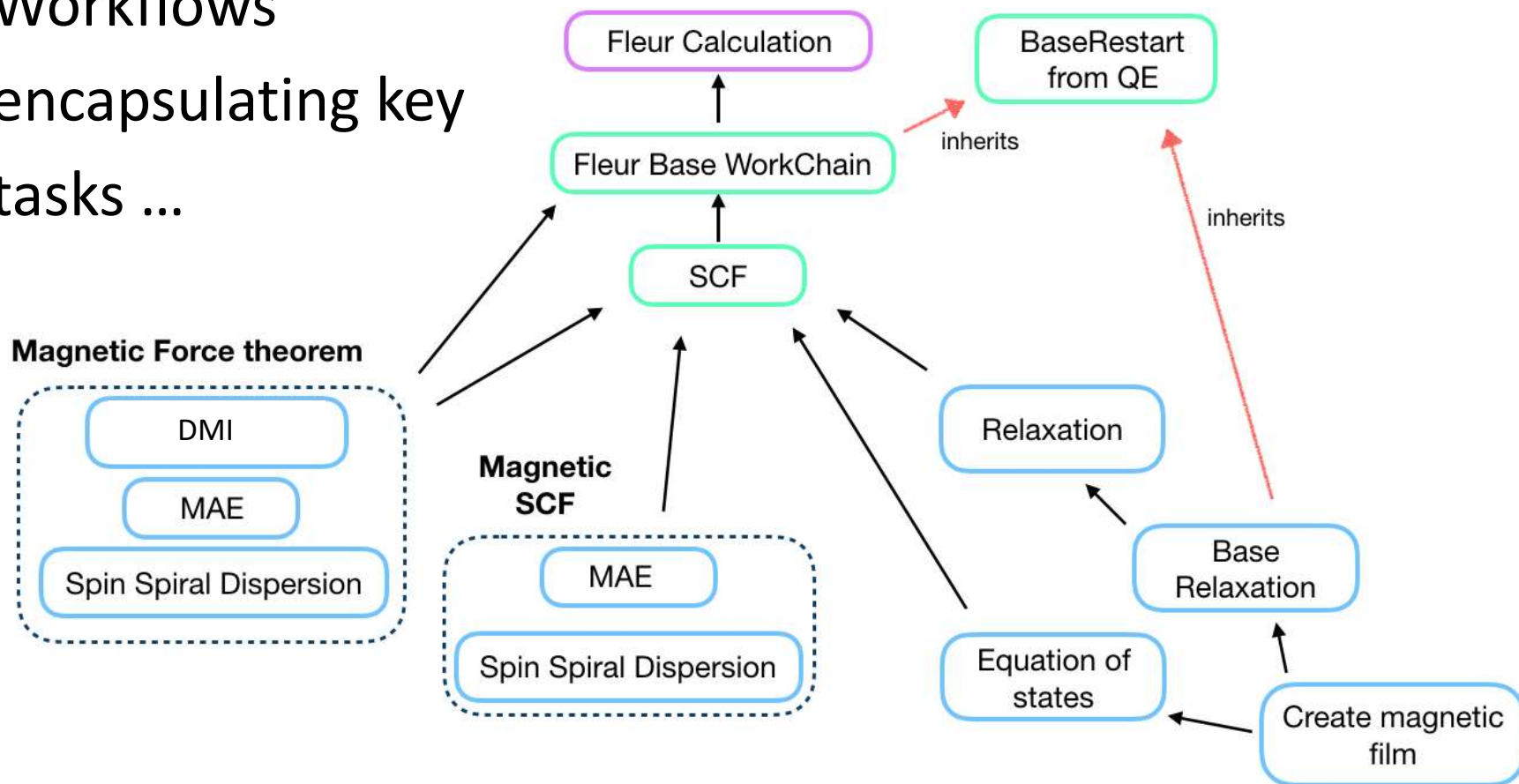
- > Download from FLEUR homepage
- > Try out using Quantum Mobile:

<https://www.materialscloud.org/work/quantum-mobile>

FLEUR AiiDA plugin provides:

- > Support for:
 - > Input generator for FLEUR-specific parameters
 - > Parsers for FLEUR input and output files
- > Specialized Data-types:
 - > For FLEUR input data
 - > For modification of FLEUR computational parameters
- > Tools for XML handling, plotting ...

Workflows
encapsulating key
tasks ...



Future development

- > Further porting and tuning of FLEUR on current and future HPC architectures
 - > More advanced implementation on GPUs
 - > Stability and interoperability with AiiDA for more complex workflows
 - > Improve user's experience
- > Hybrid functional code branch:
 - > Make available on HPC systems as well
 - > Enable it to treat many hundreds of atoms
 - > Encapsulate complex operations with products of LAPW wavefunctions for further use

Modules and libraries

- > You are a developer interested in FLEUR?
- > Key activity within MaX-COE: modularization
- > We provide stand-alone features:

LaXLIB

Generalized interface to linear algebra libraries.

In-particular to solvers for dense generalized matrix eigenvalue problems

juDFT-LIB

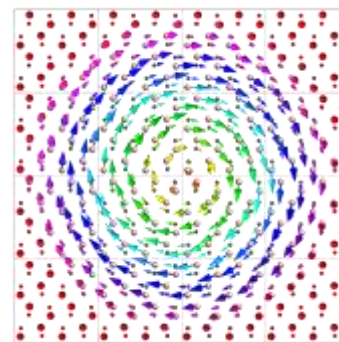
Library for common tasks used in our DFT codes:

- Timing
- Error Handling
- HDF5 IO
- ...

Summary

- > FLEUR as a versatile tool for high accuracy DFT simulations
- > High performance computing using FLEUR
- > Gimpse of the practical aspects of FLEUR usage

$$\phi = \begin{cases} e^{i(\mathbf{k}+\mathbf{G})\mathbf{r}} \\ \sum (a_L^{\mathbf{G}} u(r) + b_L^{\mathbf{G}} \dot{u}(r)) Y_L \end{cases}$$



<http://www.flapw.de>

The Flexibilities of Wavelets for Electronic Structure Calculations in Large Systems

12 November 2020
10am (CET)





DRIVING THE EXASCALE TRANSITION

Follow us on:

THANKS



company/max-centre/



[@max_center2](https://twitter.com/max_center2)



<http://www.max-centre.eu/>