MAX DRIVING THE EXASCALE TRANSITION

Using FLEUR yourself
Future of FLEUR

Daniel Wortmann

MaX "*Materials Design at the Exascale*", has received funding from the European Union's Horizon 2020 project call H2020-INFRAEDI-2018-1, grant agreement 824143

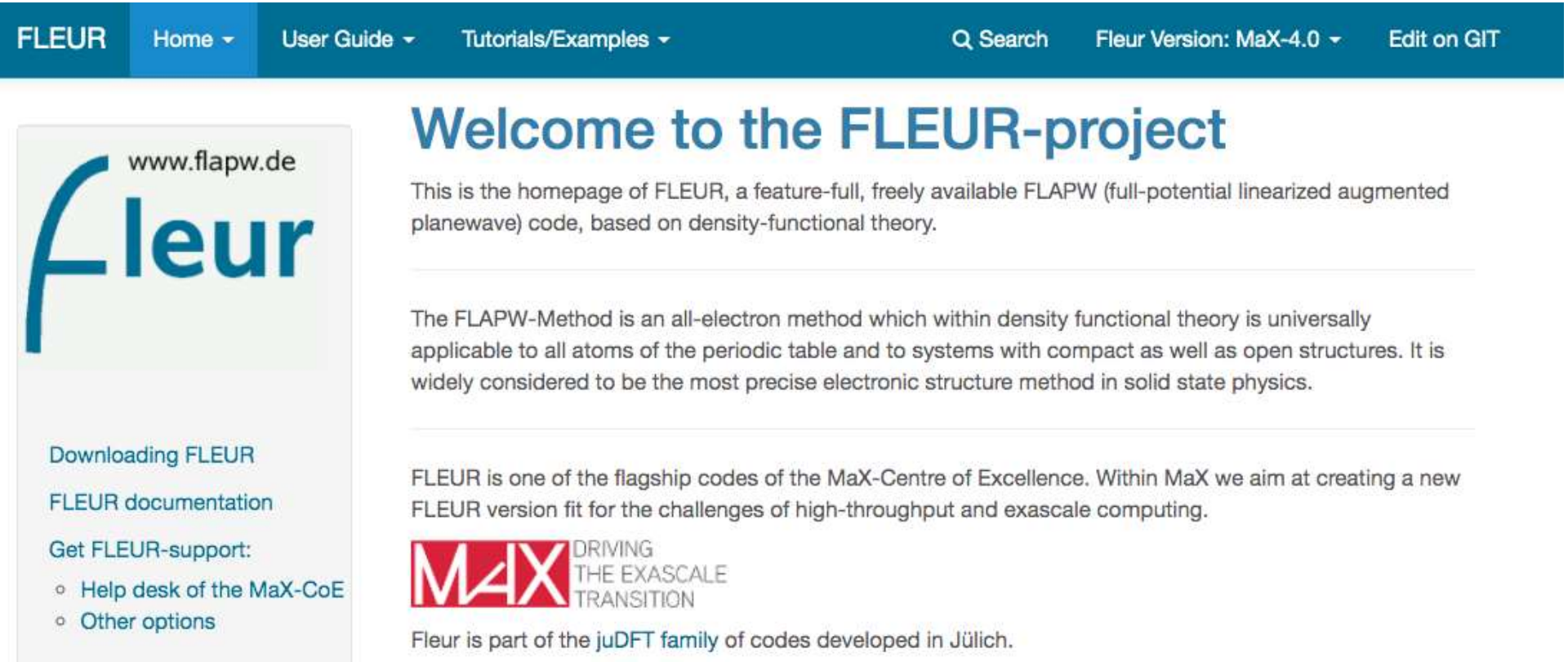
Join the FLEUR community

You are interested in:

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

Main entry point: Homepage

http://www.flapw.de



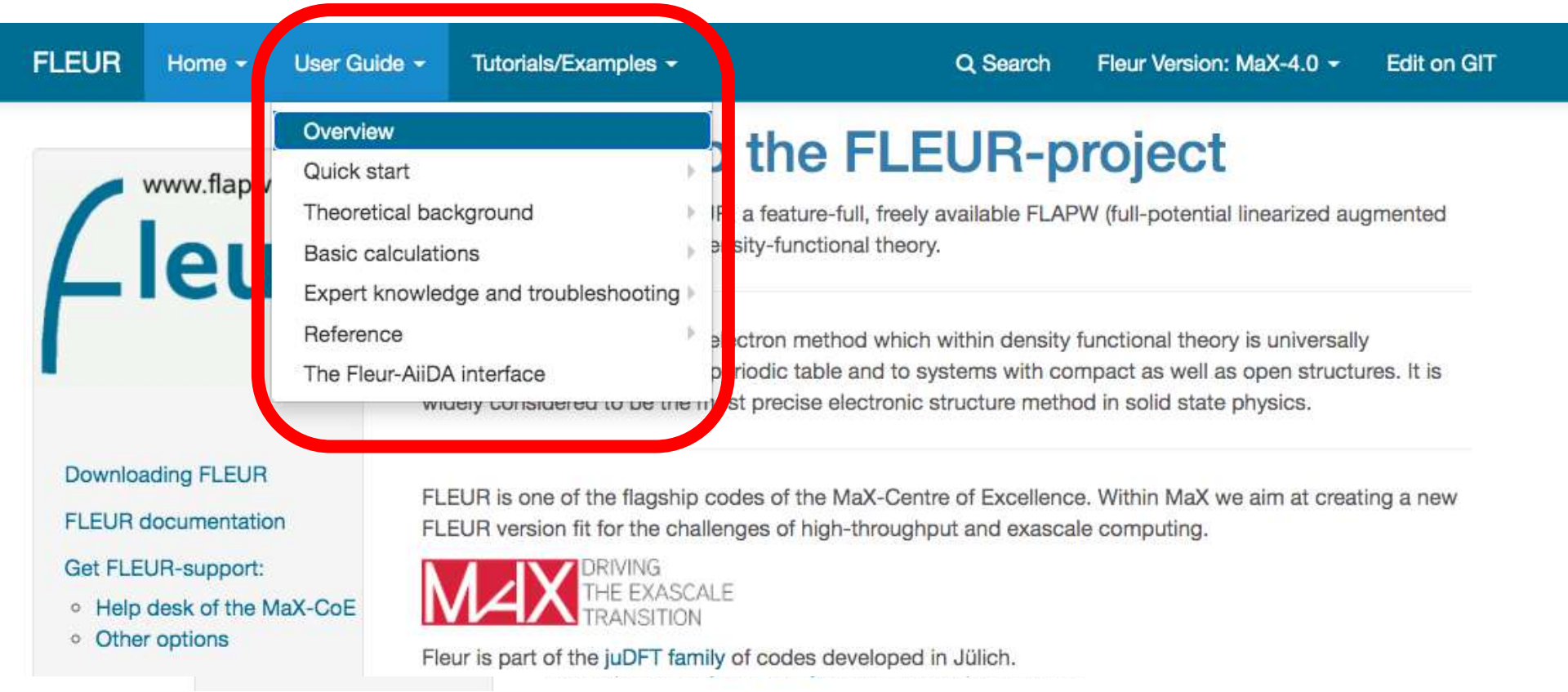
Join the FLEUR community

You are interested in:

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

Main entry point: Homepage

http://www.flapw.de



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 (peferably 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

Basic FLEUR usage

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…

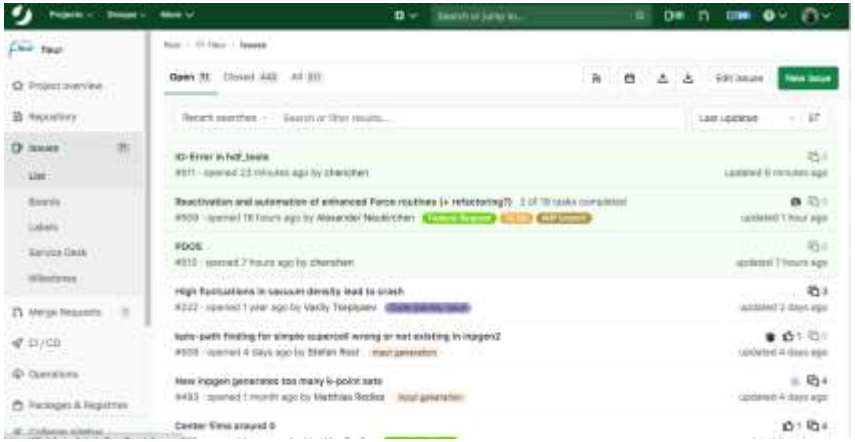
Support Channels

Documentation:
www.flapw.de

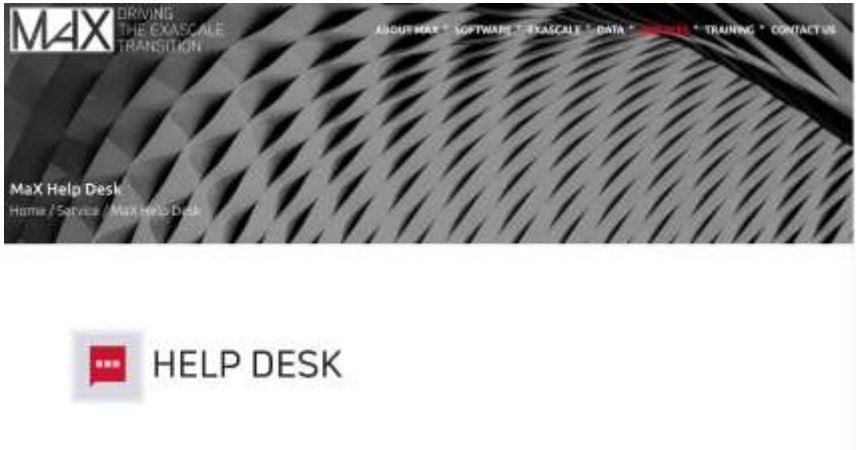
HELP needed?

FLEUR Mailing List:
fleur@fz-juelich.de

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



MaX Helpdesk:
www.max-centre.eu



FLEUR @ GPU

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

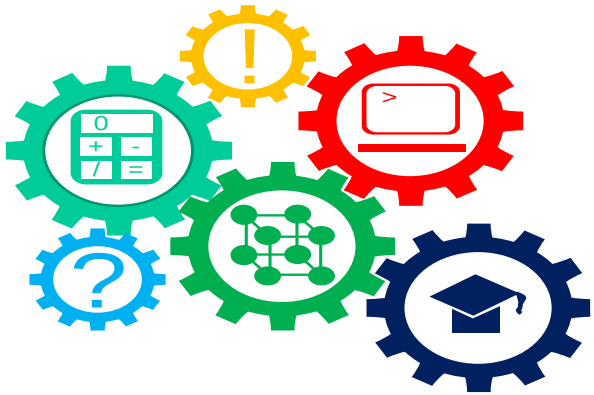
Most problematic:

- 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:

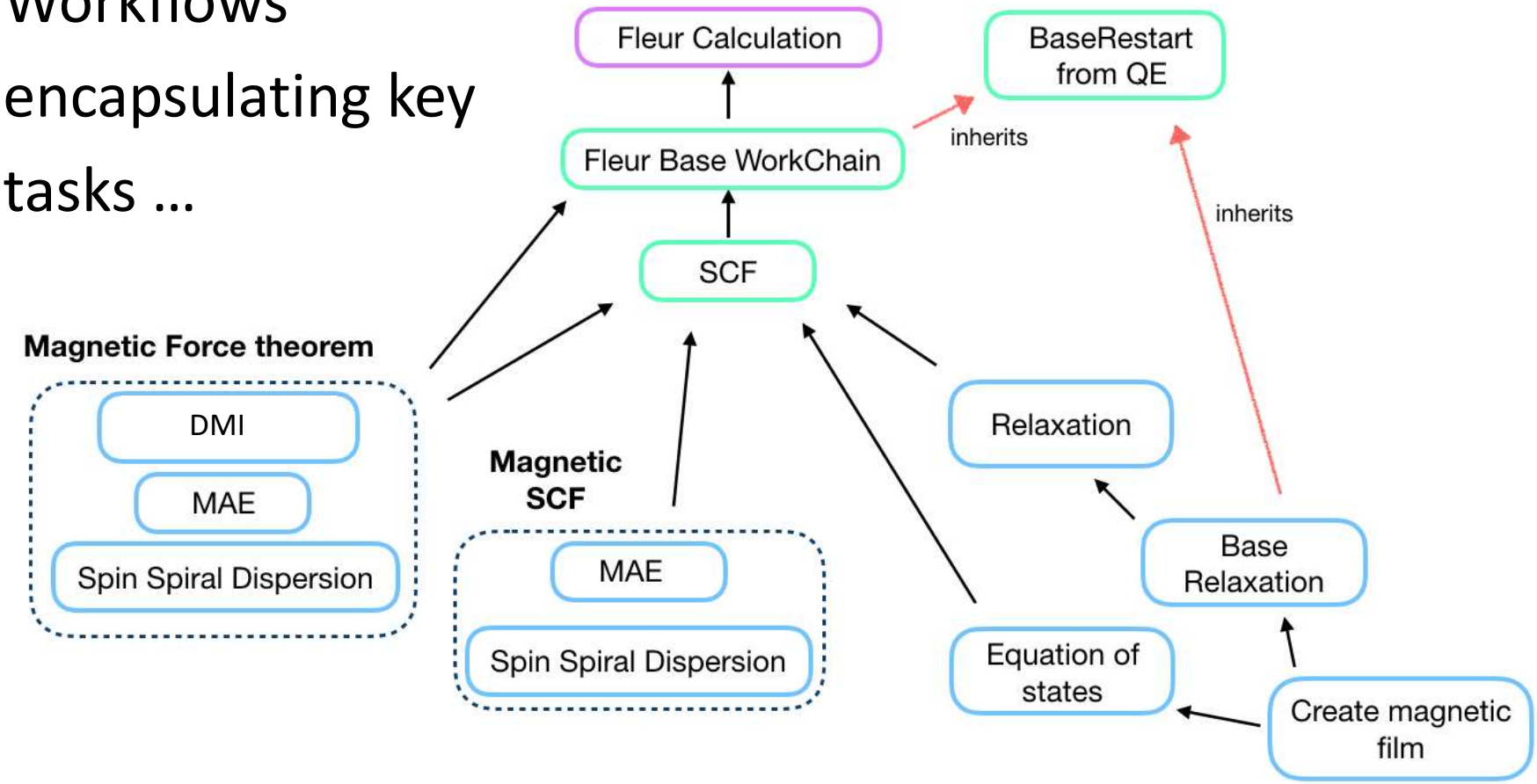
https://www.materialscloud.org/work/quantum-mobile

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 …

FLEUR-AiiDA: Workflows

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

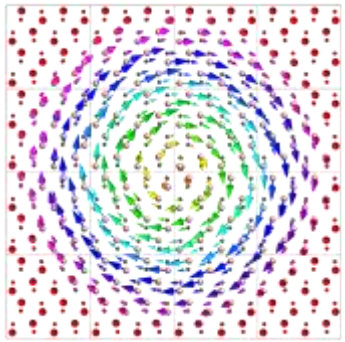
juDFT-LIB
Library for common tasks used in our DFT codes:

- 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

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



http://www.flapw.de

Next Webinar

The Flexibilities of Wavelets for Electronic Structure Calculations in Large Systems

12 November 2020

10am (CET)

DRIVING THE EXASCALE TRANSITION

THANKS

Follow us on:

company/max-centre/

@max_center2

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