

# Recent advances in DFTK.jl

Reliable black-box algorithms, performance and differentiable  
electronic-structure simulations

Jeremiah DeGreeff<sup>†</sup> and Michael F. Herbst<sup>\*</sup>

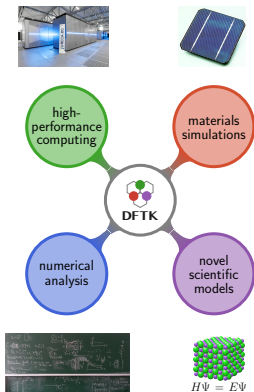
<sup>†</sup>Massachusetts Institute of Technology

<sup>\*</sup>Applied and Computational Mathematics, RWTH Aachen University

11 October 2021

Slides: [https://michael-herbst.com/talks/2021.10.11\\_dftk\\_feature\\_summary.pdf](https://michael-herbst.com/talks/2021.10.11_dftk_feature_summary.pdf)

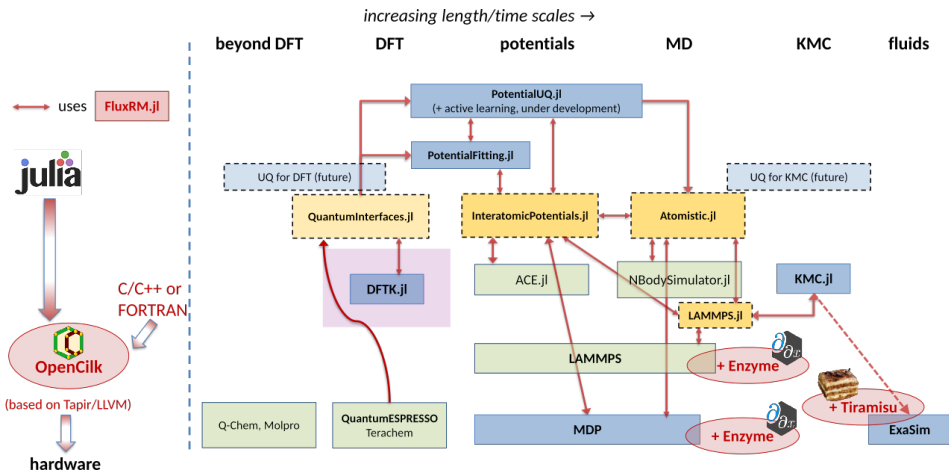
# Density-functional toolkit<sup>1</sup> — <https://dftk.org>



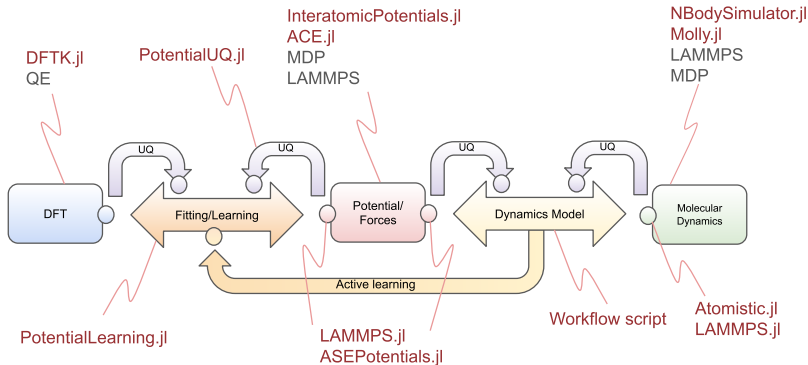
- **julia** code for density-functional theory (DFT)
  - **Fully composable** with **julia** ecosystem:
    - Arbitrary precision (32bit, >64bit, ...)
    - Automatic differentiation
    - Numerical error control
    - GPU acceleration **planned**
    - Uncertainty quantification (UQ) **planned**
  - Supports **mathematical developments** *and* scale-up to relevant **applications**
  - *i.e.* reduced problems for UQ/mathematical analysis *and* DFT on > 800 electrons
- ⇒ Build with multidisciplinary research in mind
- Avoids **two-language problem**: Just **julia**
  - Only 2.5 years of development
  - Only 6k lines of code
- ⇒ Low entrance barrier **across backgrounds**

<sup>1</sup>M. F. Herbst, A. Levitt and E. Cancès. JuliaCon Proc., 3, 69 (2021).

# Our integrated simulation approach

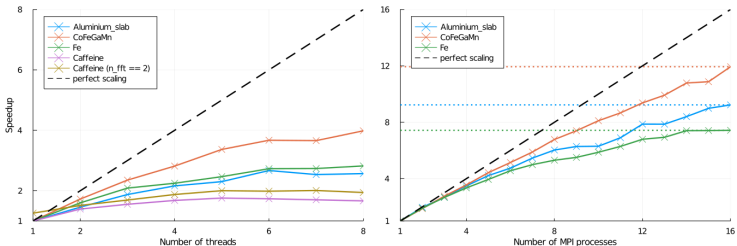


# Embedding in CESMIX composable **julia** workflow



- Provider of electronic-structure simulations (**DFT**)
- For fitting / learning *or* ab initio molecular dynamics
- Details about integration: See poster *The Atomistic Suite for CESMIX in Julia*

# DFTK status: Performance<sup>2</sup>

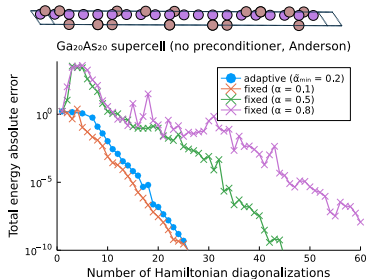
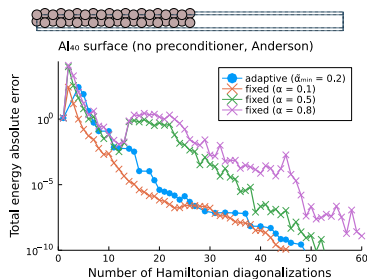


- Performance status:
  - Mixed thread-based / MPI parallelisation
  - Similar speed as established codes (small to medium problems)
  - Focus of DFTK: **Robust black-box algorithms** (next slides)
  - Little optimisation of parallel efficiency

⇒ Still: Decent strong scaling for distribution over  $k$ -points (MPI)
- Future CESMIX efforts:
  - GPU acceleration **planned**
  - Parallel efficiency **planned**

<sup>2</sup><https://docs.dftk.org/stable/guide/parallelization/>

# Black-box algorithms: Adaptive damping<sup>3</sup>



- DFT involves a fixed-point problem, solved by SCF iteration

$$\rho^{(n+1)} = \rho^{(n)} + \alpha P^{-1} \left[ \text{SCF step}(\rho^{(n)}) - \rho^{(n)} \right]$$

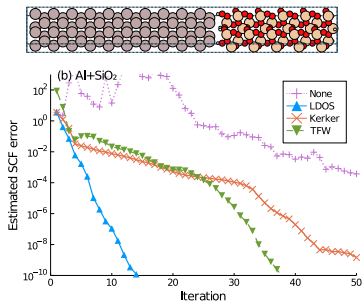
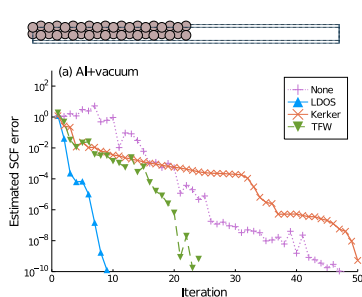
- How to choose mixing  $P^{-1}$  and damping  $\alpha$ ?
- State-of-the-art: Guessing / trial and error (**fixed damping**)

⇒ Wasted computational time!

- DFTK approach: **adaptive damping automatically** selects damping
- Similar performance than best fixed damping, but **fully black-box**

<sup>3</sup>M. F. Herbst, A. Levitt. *A robust and efficient line search for self-consistent field iterations* arXiv 2109.14018.

# Black-box algorithms: LDOS mixing<sup>5</sup>




- Long-standing problem: Suitable mixing  $P^{-1}$  for inhomogeneous systems
  - E.g. metal+insulator, like a Hafnium surface
- State-of-the-art: local Thomas-Fermi-von Weizsäcker mixing (TFW)<sup>4</sup>
- DFTK approach: LDOS mixing automatically interpolates between Kerker mixing (in the metallic region) and no mixing (insulating region)

⇒ Parameter-free and black-box

<sup>4</sup>D. Raczkowski, A. Canning, L. W. Wang, Phys. Rev. B. **64**, 121101 (2001).

<sup>5</sup>M. F. Herbst, A. Levitt. J. Phys. Condens. Matter **33**, 085503 (2021).

# Algorithmic differentiation (AD) in DFTK

- Early stage: Development started in June by a GSoC student
- Status: All required building blocks implemented
- Forward-mode AD working, reverse-mode is work-in-progress
- Simple example: Stresses ( $\approx$  sensitivity energy wrt. lattice)
- Formal definition vs.  code<sup>6</sup>

$$\text{Stress} = \frac{1}{\det(\mathbf{L})} \left. \frac{\partial E_{\text{DFT}}[P_*, (\mathbf{I} + \mathbf{M}) \mathbf{L}]}{\partial \mathbf{M}} \right|_{\mathbf{M}=\mathbf{0}}$$

```
scfres = self_consistent_field(basis) # Run SCF, get P*  
L = basis.model.lattice  
stress = 1/det(L) * ForwardDiff.gradient(M -> recompute_energy(scfres, (I + M) * L),  
                                          zero(L))
```

- Outlook:
  - Sensitivities: Structural, alchemical, model parameters
  - Intrusive analysis of DFT uncertainties

<sup>6</sup>Live code: <https://github.com/JuliaMolSim/DFTK.jl/blob/master/src/postprocess/stresses.jl>



# Funding and Acknowledgements

- Eric Cancès (ENPC ParisTech)
- Antoine Levitt (Inria Paris)
- Niklas Schmitz (TU Berlin)
- Benjamin Stamm (RWTH Aachen)



Summer of code





Applied and  
Computational  
Mathematics




**RWTH**AACHEN  
UNIVERSITY



# Contact information

 **DFTK** <https://dftk.org>

 jrdegreeff  
 degreeff@mit.edu

 mfherbst  
 <https://michael-herbst.com/blog>  
 herbst@acom.rwth-aachen.de