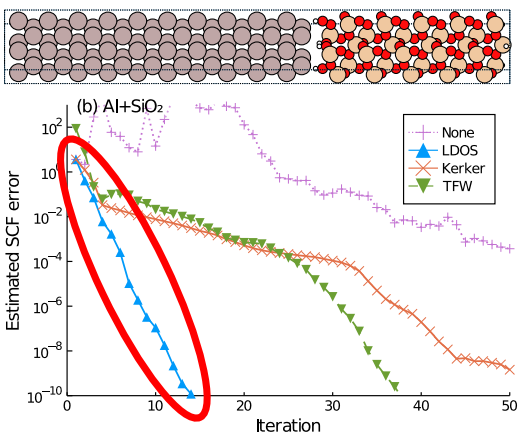
Recent updates in DFTK

Michael F. Herbst

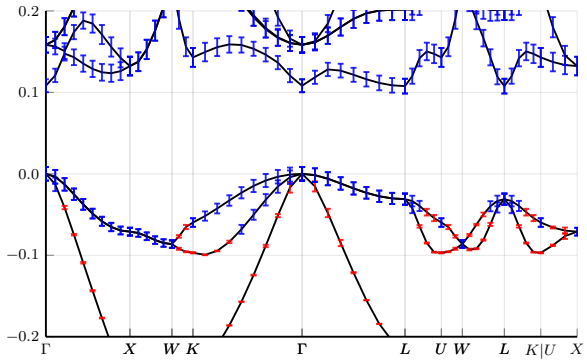
Applied and Computational Mathematics, RWTH Aachen University

22 July 2022

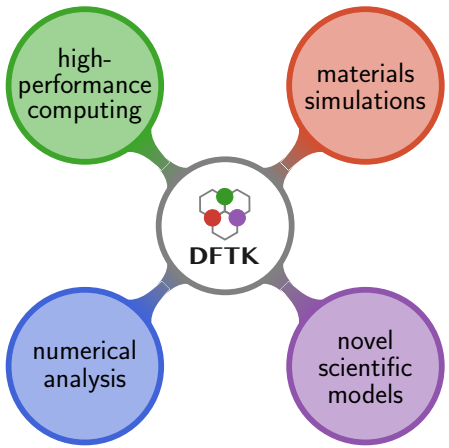
Slides: `https://michael-herbst.com/talks/2022.07.22_dftk_juliacon.pdf`



Self-adapting black-box DFT algorithms

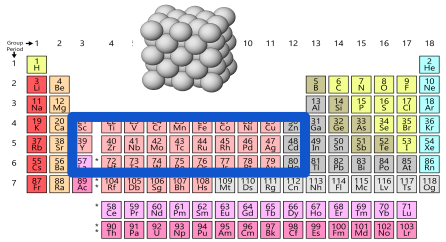


Error control of DFT quantities

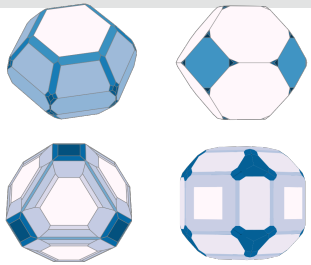


Contents

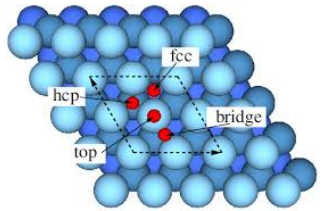
Motivation: Computational materials design[1]



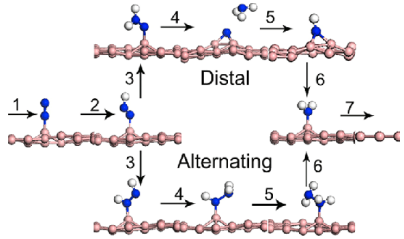
Host metal + dopant
$\simeq 30 \times 30 = 900$



Host surface
$\simeq 3 - 5$



Dopant adsorpotion site
$\simeq 30$

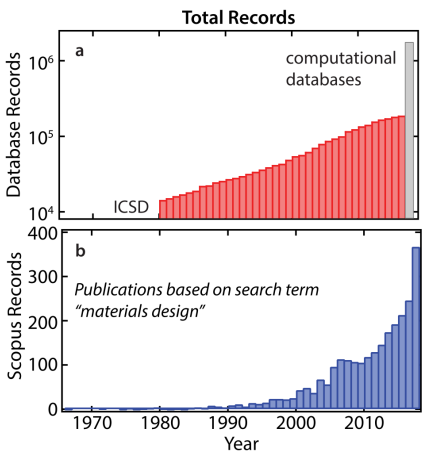


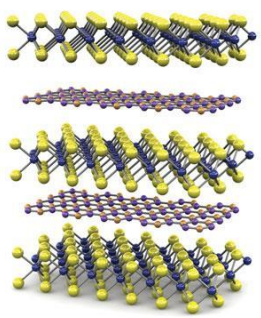
Reaction intermediates
$\simeq 10$

- Simple catalyst design space: Combinatorial $\simeq 10^5 - 10^6$ possibilities
- Computational screening to complement and accelerate experiment
- Exponentially growing impact & successes:[5]
  - Semiconductors[2], batteries[3], magnetic compounds[4]

$\Rightarrow$ Crucial tool for 21st century challenges:

- Information technology & quantum computing
- Energy storage & conversion





[1] ACED project: `https://www.cmu.edu/aced/index.html`

[2] S. Luo *et. al.* WIREs Comput. Mol. Sci. **11**, e1489 (2021).

[3] L. Kahle *et. al.* Energy & Environ. Science, **13**, 928 (2020).

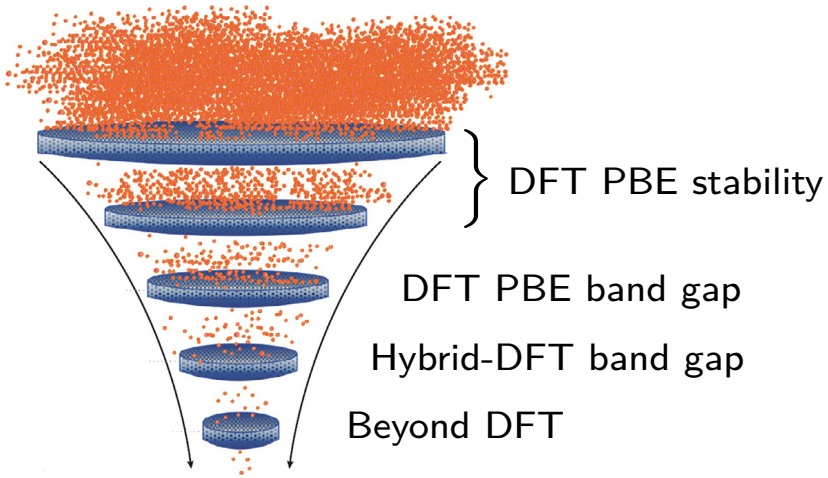
[4] S. Jiang *et. al.* J. Alloys Comp. **867**, 158854 (2021).

[5] K. Alberi *et. al.* J. Phys. D, **52**, 013001 (2019).

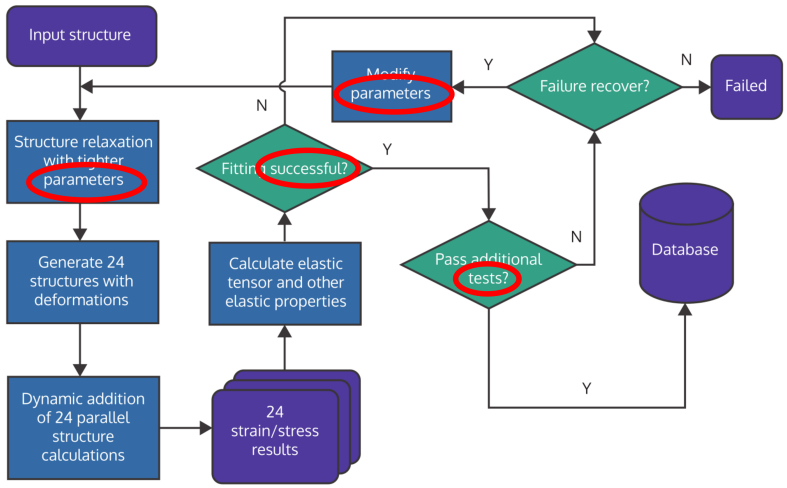
`https://www.edfenergy.com/electric-cars/batteries`

J. Evans *Beyond graphene* Chemistry World (2014).

Sketch of high-throughput workflows



Design funnel for photovoltaic materials



Workflow for computing elasticity tensors

- Many parameters to choose (algorithms, tolerances, models)
  - Elaborate heuristics: Failure rate $\simeq 1\%$
  - Still: Thousands of failed calculations

  $\Rightarrow$ Wasted resources & increased human attention (limits througput)
- Carbon footprint? More complex design spaces?
- **Goal:** Self-adapting black-box DFT algorithms
  - Parameter-free, automatically adapt to simulated system
  - Transform empirical wisdom to built-in convergence guarantees

G. Hautier Comput. Mater. Sci. **164**, 108 (2019); L. Himanen *et. al.* Adv. Science **6**, 1900808 (2019).

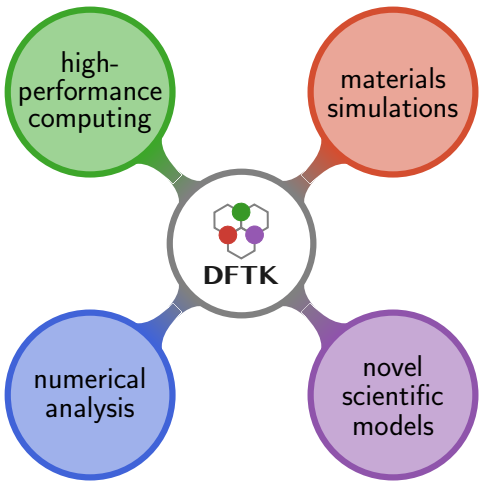
Interdisciplinary field ⇒ Multidisciplinary community

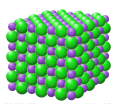
- **Mathematicians:** Toy models and unphysical edge cases
- **High-performance person:** Exploit hardware specialities
- **Scientist:** Design new models, not tweak numerics
- **Practitioner:** Reliable, black-box code, high-level interface

- State-of-the-art first-principle codes:
  - Difficult problem ⇒ Complex codes
  - Hard-coded details: Workflow, algorithms, optimisations
  - Huge code bases: 1M lines and beyond
  - Non-standard input syntax and API
  - Two-language problem: Algorithmic code hardly accessible

⇒ Innovations might not cross community boundaries

Density-functional toolkit[1] — https://dftk.org





$H\Psi = E\Psi$

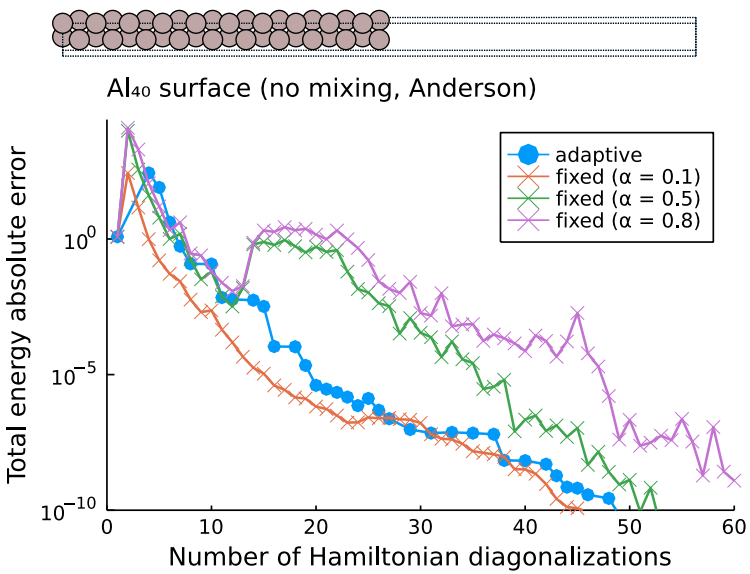
- julia code for plane-wave DFT, started in 2019
- Lowers barriers for cross-disciplinary research:
  - Allows restriction to relevant model problems,
  - scale-up to application regime (1000 electrons)
  - Features incl. meta-GGA, response, MPI
  - Speed within factor 2–4 to established codes

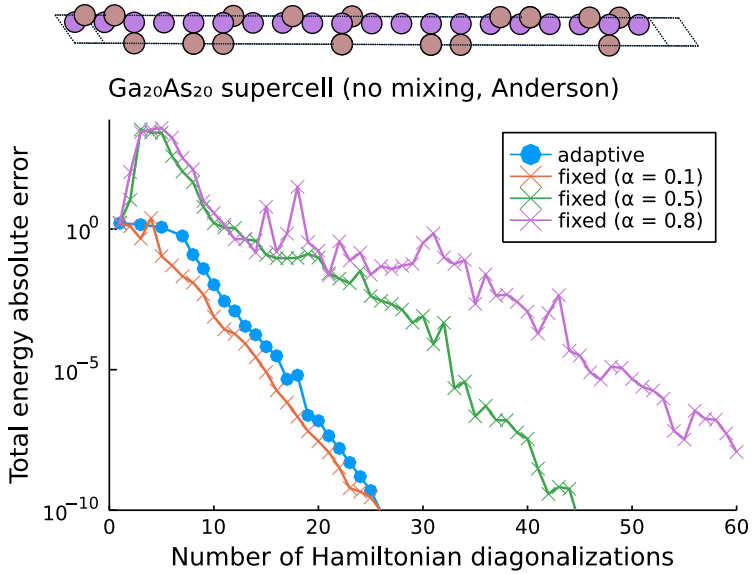
$\Rightarrow$ Build to enable multidisciplinary synergies

- Fully composable with julia ecosystem:
  - Arbitrary precision (32bit, >64bit, . . .)
  - Algorithmic differentiation (AD)
  - Integrated in multi-scale pipeline
    (potentials, MD, see talk by Emmanuel Lujan)
- High-productivity research framework:
  - Only 7000 lines of code
  - Avoids two-language problem: Just julia
  - GSoC student (10 weeks) for initial AD support

Contents

Black-box algorithms: Adaptive damping[1]





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

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

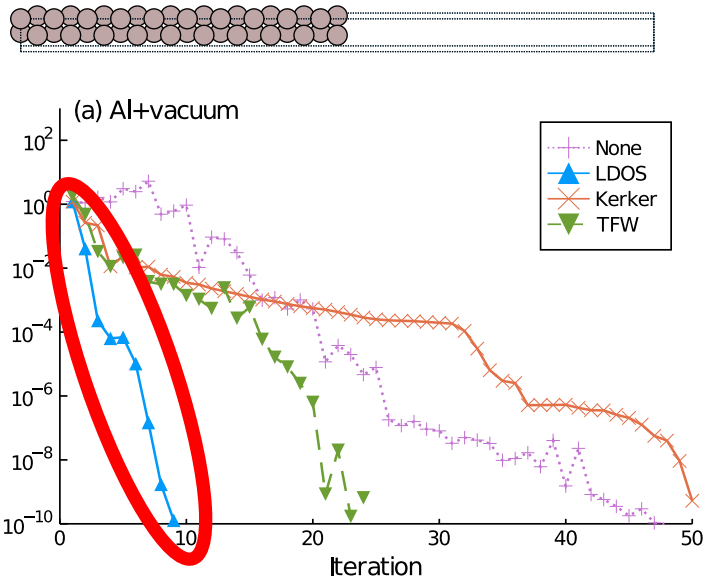
- How to choose mixing P^{-1} and damping α?
- State-of-the-art: Guessing / trial and error (fixed damping)

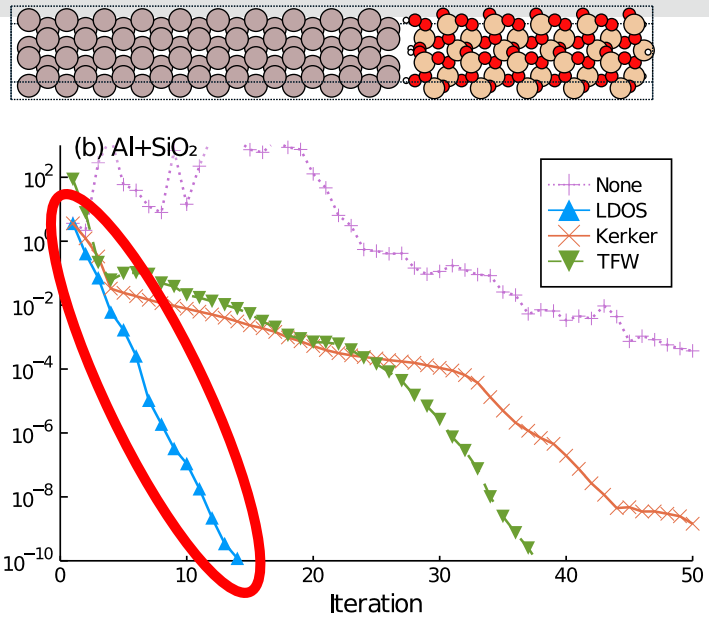
$\Rightarrow$ Wasted computational time!

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

[1] M. F. Herbst, A. Levitt. *A robust and efficient line search for self-consistent field iterations* arXiv 2109.14018.

Black-box algorithms: LDOS mixing[2]





- Long-standing problem: Suitable mixing for inhomogeneous systems
  - E.g. metal+insulator, catalytic surfaces, ...
- State-of-the-art: local Thomas-Fermi-von Weizsäcker mixing (TFW)[1]
- **DFTK** approach: LDOS mixing *automatically* interpolates between Kerker mixing (in the metallic region) and no mixing (insulating region)

$\Rightarrow$ Parameter-free and black-box

---

[1] D. Raczkowski, A. Canning, L. W. Wang, Phys. Rev. B. **64**, 121101 (2001).

[2] M. F. Herbst, A. Levitt. J. Phys. Condens. Matter **33**, 085503 (2021).

Obtaining DFT model sensitivities

- Consider optimal lattice constant

$$a_* = \arg\min_a \mathcal{E}_{\mathrm{DFT}}(a, \theta)$$

(a lattice parameter, θ DFT exchange-correlation parameters)

- How sensitive is a_* for a system? $\Rightarrow$ Need $\frac{da_*}{d\theta}$
- Annoyances for derivation and implementation:
  - Nested iterative methods (eigensolver, SCF, lattice optimisation)
  - Unusual second-order derivatives (e.g. $\frac{\partial^2 \mathcal{E}}{\partial \theta \partial a}$)
  - Combinatorial explosion: Other QoI? Other DFT models?
  - What about future DFT models?
    (with completely different kinds of parameters θ)
- Use algorithmic differentiation ($\approx$ automatic derivatives)
  - Break "one PhD student per derivative" paradigm
  - Generic framework for DFT derivatives / response properties

  $\Rightarrow$ New properties/derivatives by non-DFT experts!

Example: Lattice constant sensitivities of silicon

```
function dft_energy(a, θ)
    model = model_DFT(make_structure(a), PbeExchange(θ))
    basis = PlaneWaveBasis(model; Ecut=..., kgrid=... )
    self_consistent_field(basis).energies.total
end
optimise_lattice(θ) = optimise(a -> dft_energy(a, θ))

sensitivities =
    ForwardDiff.gradient(optimise_lattice, [κ, β])
```

$$a_* = \arg\min_a \mathcal{E}(a, \theta)$$

$$\text{sensitivities} = \frac{da_*}{d\theta}$$

- One line of code to "request" desired gradients from:
  - Floating-point agnostic DFTK architecture
  - julia tools for algorithmic differentiation
  - Generic response implementation in DFTK
- Fully flexible in DFT model or targeted quantity:
  - Only XC energy expression & SCF postprocessing code needed

(Å)	a_*	κ	$\frac{da_*}{d\kappa}$	β	$\frac{da_*}{d\beta}$
expmnt.	5.421				
PBEsol	5.449	0.804	0.713	0.0375	0.0058
PBE	5.461	0.804	0.550	0.0667	0.0194
APBE	5.465	0.804	0.482	0.0790	0.0269
PBEmol	5.467	0.804	0.456	0.0838	0.0301
XPBE	5.466	0.920	0.603	0.0706	0.0184
rev-PBE	5.467	1.245	0.744	0.0667	0.0099

Model sensitivities for the silicon lattice constant

DEMO

Ecosystem integration of DFTK

- `AtomsBase.jl`: Interface for atomic structures
  - `AtomIO.jl`: Abstraction for parsing/writing structure files (ongoing student project)
- `DftFunctionals.jl`: Interface and differentiable fallbacks for exchange-correlation functionals
- `InteratomicPotentials.jl`: Interface for evaluating energies and forces

Contents

Ongoing developments

- GPU Support (GSoC of Guillaume Vigne):
  - **Goal:** Minimise code duplication between GPU & CPU
  - Target generic `GpuArrays` infrastructure $\Rightarrow$ CUDA & AMD

- Adjoint-mode algorithmic differentiation:
  - (talk by Gaspard, Markus, Niklas Wed 27th 12:30 UTC)
  - All parameter sensitivities by a single response problem

  $\Rightarrow$ Routine computation of model sensitivities

  $\Rightarrow$ Prospect: Machine-learned solid-state XC models

DFTK: Bringing mathematical research to the applications

- Self-adapting black-box DFT methods[a,b] (details follow)
- Numerical analysis of DFT[c]
- Practical numerical error bounds[d,e]

- Exploring algorithmic differentiation: DFT, potentials, MD
- Uncertainty quantification all the way: DFT, potentials, MD
- Approximate computing on modern GPUs

- Outreach and teaching
  - Community building: julia-based first-principle ecosystem
  - Interdisciplinary workshop on DFT and DFTK
  - Lecture: Mathematics of computational chemistry

Growing user base:

MIT
KU The University of Kansas
ASU Arizona State University
Carnegie Mellon University
École des Ponts ParisTech
Inria
UBFC Université Bourgogne Franche-Comté
서울대학교 Seoul National University
RWTH Aachen University
TUM

[a] MFH, A. Levitt. J. Phys. Condens. Matter **33**, 085503 (2021).

[b] MFH, A. Levitt. J. Comput. Phys. **459**, 111127 (2022).

[c] E. Cancès, G. Kemlin *et. al.* J. Matrix Anal. Appl., **42**, 243 (2021).

[d] MFH, A. Levitt, E. Cancès. Faraday Discus. **223**, 227 (2020).

[e] E. Cancès, G. Dusson *et. al.* arxiv 2111.01470v1.

Acknowledgements

https://michael-herbst.com/talks/2022.07.22_dftk_juliacon.pdf

École des Ponts
Antoine Levitt
Eric Cancès

RWTH
Benjamin Stamm
Markus Towara

Mines Paris
Guillaume Vigne

MIT
Valentin Churavy
Jeremiah DeGreeff
Emmanuel Luján

TU Berlin
Niklas Schmitz

DFTK contributors

Summer of code

Questions?

https://michael-herbst.com/talks/2022.07.22_dftk_juliacon.pdf


mfherbst

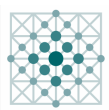

herbst@acom.rwth-aachen.de


https://michael-herbst.com/blog


DFTK https://dftk.org

JuliaCon 2021 *A mathematical look at electronic structure theory*

JuliaCon 2022 *Automatic Differentiation for Quantum Electron Structure* (Wed 27th July 12:30 UTC)



RWTH AACHEN UNIVERSITY