DFTK.jl: A multidisciplinary Julia code for density-functional theory development

Michael F. Herbst

CERMICS, Inria Paris and École des Ponts ParisTech
https://michael-herbst.com

3rd February 2021

→ https://michael-herbst.com/talks/2021.02.03_dftk_tst_cesmix.pdf

Contents

Why is electronic structure theory so hard?

- Regime of quantum mechanics
- System: Hamiltonian $\hat{\mathcal{H}}$, differential operator
- Minimisation problem: Ground state Ψ with energy

$$E = \min_{\Psi} \int_{\mathbb{R}^{3N}} \Psi\left(\underline{\boldsymbol{r}}_1, \underline{\boldsymbol{r}}_2, \ldots, \underline{\boldsymbol{r}}_N\right) \hat{\mathcal{H}} \Psi\left(\underline{\boldsymbol{r}}_1, \underline{\boldsymbol{r}}_2, \ldots, \underline{\boldsymbol{r}}_N\right) \mathrm{d}\underline{\boldsymbol{r}}_1 \cdots \mathrm{d}\underline{\boldsymbol{r}}_N$$

- Electronic properties: Derivatives of the energy

- Challenge: Size of N, e.g. 2 silicon atoms: $N = 28$
- 2 quadrature points per DOF $\Rightarrow \mathbf{2}^{84} \approx 2 \cdot 10^{25}$ integrand evals

$\Rightarrow$ Finished in 1 year:

Why is electronic structure theory so hard?

- Regime of quantum mechanics
- System: Hamiltonian $\hat{\mathcal{H}}$, differential operator
- Minimisation problem: Ground state Ψ with energy

$$E = \min_{\Psi} \int_{\mathbb{R}^{3N}} \Psi\left(\underline{\boldsymbol{r}}_1, \underline{\boldsymbol{r}}_2, \ldots, \underline{\boldsymbol{r}}_N\right) \hat{\mathcal{H}} \Psi\left(\underline{\boldsymbol{r}}_1, \underline{\boldsymbol{r}}_2, \ldots, \underline{\boldsymbol{r}}_N\right) \mathrm{d}\underline{\boldsymbol{r}}_1 \cdots \mathrm{d}\underline{\boldsymbol{r}}_N$$

- Electronic properties: Derivatives of the energy

- Challenge: Size of N, e.g. 2 silicon atoms: $N = 28$
- 2 quadrature points per DOF $\Rightarrow \mathbf{2}^{84} \approx 2 \cdot 10^{25}$ integrand evals

$\Rightarrow$ Finished in 1 year: ≈ 1.5 attoseconds per eval

Density-functional theory

$\Rightarrow$ Brute force won't cut it

$\Rightarrow$ Cost / quality balance of approximate models ($\simeq 10^3 - 10^4$)

$\Rightarrow$ E.g. density-functional theory (DFT) approximation

- Amongst the most-used family of models
- Effective one-particle model ($N = 1$)
- May construct DFT model for specific context
- Discretisation basis: Build known physics into model
- But: Non-convex, non-linear minimisation

Self-consistent field procedure

- Euler-Lagrange equations (DFT):

$$\begin{cases} \hat{\mathcal{F}}_\rho = -\frac{1}{2}\Delta + V_{\boldsymbol{\rho}} \\ \boldsymbol{\rho}(\underline{\boldsymbol{r}}) = \sum_i f_{\varepsilon_F}(\varepsilon_i)\,|\psi_i(\underline{\boldsymbol{r}})|^2 \quad \text{with } \hat{\mathcal{F}}_\rho\psi_i = \varepsilon_i\psi_i, \\ \qquad \varepsilon_F \text{ chosen such that } \int \rho\,\mathrm{d}\underline{\boldsymbol{r}} = N, \\ \qquad \text{and } f_{\varepsilon_F}(x) = \left[1 + \exp\left(\frac{x-\varepsilon_F}{T}\right)\right]^{-1} \end{cases}$$

- Self-consistent field procedure (SCF):
  (1) Guess initial density $\boldsymbol{\rho}$
  (2) Build Kohn-Sham operator $\hat{\mathcal{F}}_\rho$
  (3) Diagonalise it to get new $\{\psi_i\}_i$
  (4) Build new $\boldsymbol{\rho}$ go to (2).

Obstacles for (high-throughput) DFT calculations

- SCF requires nested layers of solvers:
  - Eigenproblem inside fixed-point problem
  - Algorithms? Preconditioning? Tolerances?
- Accuracy-related parameters chosen by experience
  - Even in automated workflows![1]

⇒ Empirical balance: Accuracy *versus* speed *versus* reliability

- Promising aspects, hardly explored:
  - Numerical analysis to reduce number of parameters
  - More specific preconditioners to capture physics
  - Precision reduction, GPGPU, other accelerators

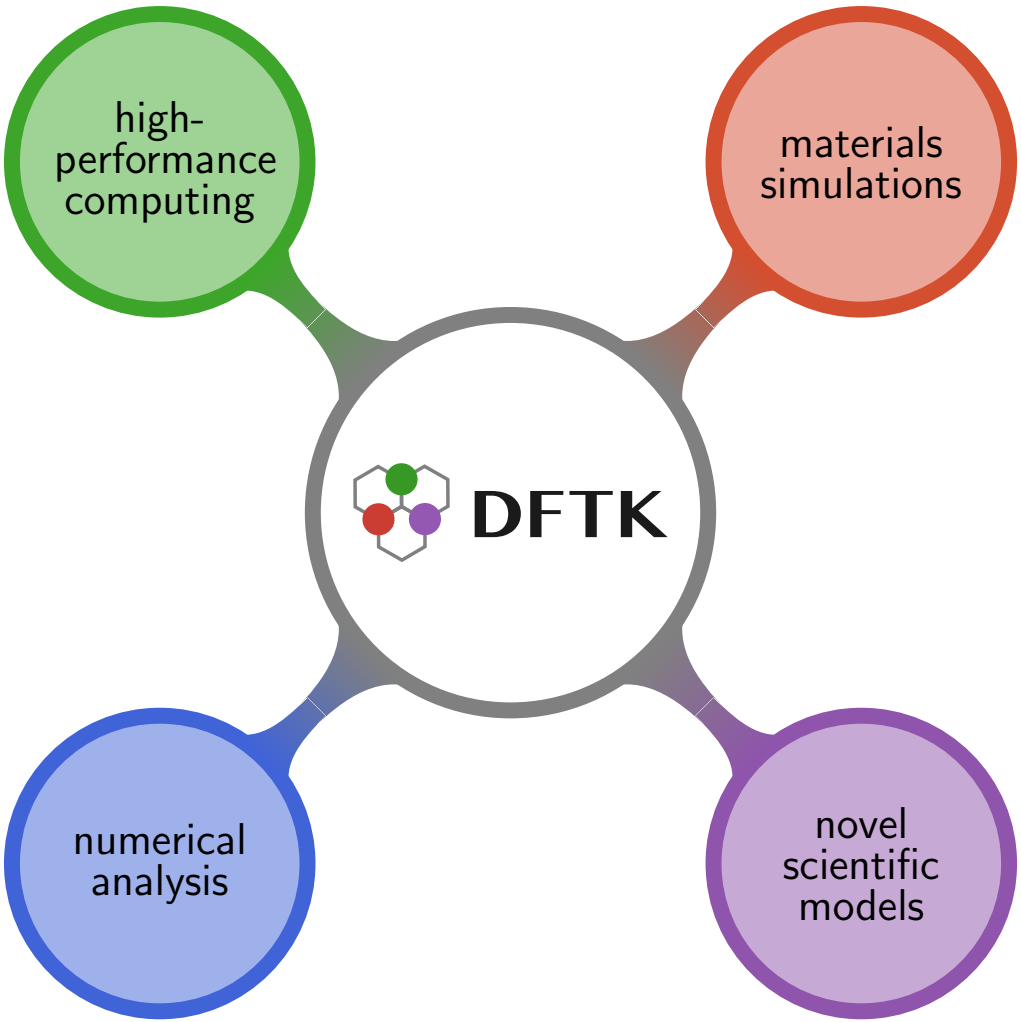
[1] L. Chanussot *et. al.* The Open Catalyst 2020 (OC20) Dataset, 2020, arXiv 2010.09990.

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 DFT codes:
  - Difficult problem ⇒ Complex codes
  - Hard-coded: Workflow / algorithms / hardware optimisations
  - Huge code bases (1M lines and beyond)
  - Non-standard input syntax and API
  - Two-language problem: Where to cut?

Density-functional toolkit (DFTK)



DFTK — https://dftk.org

- < 2 years of development, ≈ 6000 lines of julia
- Sizeable feature list (see `https://docs.dftk.org`):
  - Ground state and a bit of response theory
  - Multitude of SCF approaches (> 800 electrons possible)
  - Compose your model (e.g. analytic potentials, . . .)
  - 1D / 2D / 3D systems
  - Arbitrary floating point type
  - Mixed MPI-Thread-based parallelism
  - Integration with materials-related python modules
- Performance: Within factor 2 of established codes
- Documentation and examples: `https://docs.dftk.org`

Why julia?

Walks like Python, talks like Lisp, runs like FORTRAN

- Rich ecosystem (Optimisation, PDEs, stochastic processes, GPUs, Machine-Learning, statistics, linear algebra . . .)
- High-level, compiled and hackable
- No two-language problem: Everything stays within julia
- Multiple dispatch:
  - Generic fallbacks, fast code for special cases

  $\Rightarrow$ First get it to work then get it to work *fast*

  $\Rightarrow$ Write code once, re-use for many data structures / back ends
- `https://michael-herbst.com/learn-julia`

Current research with DFTK

⇒ Vision: Improve high-throughput workflows:

- Use physics: Reliable black-box SCF algorithms[1]
- Use maths: Error estimates and automatic error balancing[2]
- Better algorithms: Numerical analysis of SCF methods[3]

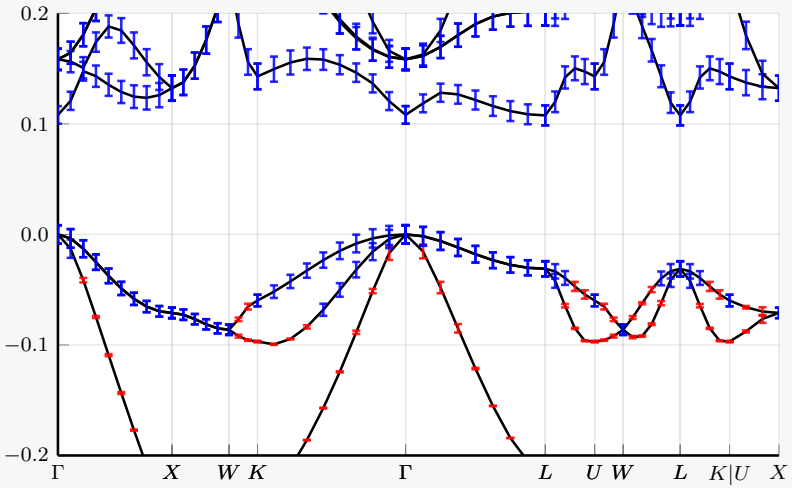
⇒ Reduction of parameters in DFT workflows

- Key unique features of DFTK:
  - Support mathematical developments *and* scale-up to regime relevant to applications
  - Low entrance barrier for researchers

[1] M. F. Herbst, A. Levitt and E. Cancès. Faraday Discuss. **224**, 227 (2020).
[2] M. F. Herbst, A. Levitt. J. Phys. Condens. Matter **33**, 085503 (2021).
[3] E. Cancès, G. Kemlin, A. Levitt. arXiv 2004.09088 (2020).

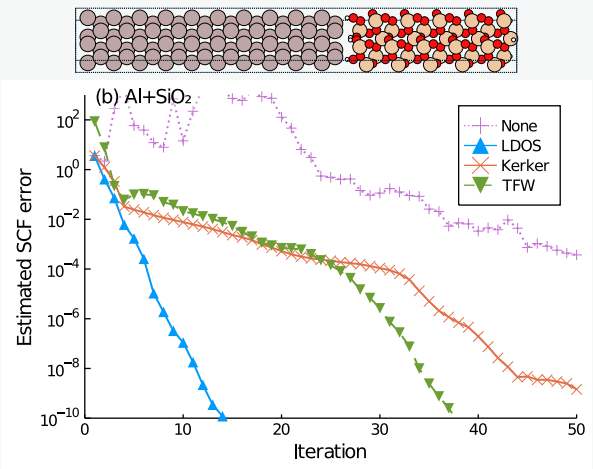
Highlighted DFTK projects

Error estimates for Kohn-Sham[1]



- A posteriori estimates for non-self-consistent Kohn-Sham
- Estimation of basis error, diagonalisation error, arithmetic error
- Time to publication: 10 weeks

SCF for inhomogeneous systems[2]



- SCFs for large inhomogeneous materials hard to converge
- LDOS: Black-box and parameter-free mixing scheme
- Same implementation for initial exploration and > 800 electrons

[1] M. F. Herbst, A. Levitt and E. Cancès. Faraday Discuss. **224**, 227 (2020).
[2] M. F. Herbst, A. Levitt. J. Phys. Condens. Matter **33**, 085503 (2021).

Outlook

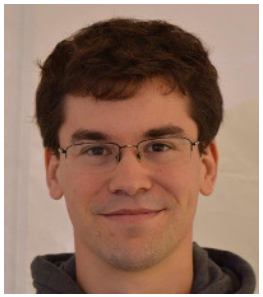
- *a posteriori* error estimates for DFT
  - Adaptive DFT methods with guaranteed error certification
  - Complements uncertainty quantification e.g. in ML, AIMD
  - \+ Reliability
- Improved SCF algorithms
  - Reduce parameters (for damping, mixing, preconditioners)
  - But: Use physics and maths, not heuristics
  - \+ Practicability for high-throughput
- Acceleration using GPUs
  - Challenge: Keep flexibility in the code
  - Wide range of suitable tools in Julia
  - \+ Speed

DEMO time

DEMO

Using  **DFTK** in practice

Acknowledgements



Antoine Levitt

Eric Cancès

Inria

École des Ponts ParisTech

SORBONNE UNIVERSITÉ

European Research Council
Established by the European Commission

julia

Other DFTK contributors:
E. Berquist, G. Kemlin, L. Ponet, D. Santra, S. Siraj-Dine, Z. Tóth

Questions?

https://dftk.org

https://michael-herbst.com/learn-julia

mfherbst

https://michael-herbst.com/blog

michael.herbst@inria.fr