High-throughput density-functional theory:

An interdisciplinary challenge

Michael F. Herbst, Antoine Levitt, Eric Cancès

CERMICS, Inria Paris and École des Ponts ParisTech

17th December 2020

→ https://michael-herbst.com/talks/2020.12.17_scientific_computing_kl.pdf

Contents

Contents

MATHerials Inria École des Ponts ParisTech

Electronic structure theory

- Modelling and understanding behaviour of electrons in matter

Electronic structure theory

- Modelling and understanding behaviour of electrons in matter

Materials and semiconductors

CC-by-3.0 https://en.wikipedia.org/wiki/File:Silicon.jpg

Chemical and pharmaceutical industry

CC-by-1.0 https://en.wikipedia.org/wiki/File:Lab_bench.jpg

Domains and societal challenges

- Renewable energies
- Catalyst design
  - Fertilisation, cleaner chemical processes, ...
- Battery materials
  - Electric cars, energy storage, ...
- IT materials
  - Usage of topological materials, spintronics, twisttronics, ...
- Drug discovery
- ...

Aren't experiments good enough?

- Experiments are expensive (money, people, time)
- 1 droplet water[1]: $1.7 \cdot 10^{21}$ particles
- Experiments only measure averages
- Sometimes hard to link to physical laws

$\Rightarrow$ Cooperative research of experiment and theory

$\Rightarrow$ Standard practice in industry and research

[1] Assume $0.05\,\mathrm{ml}$.

The need for high-throughput

- Design space is absolutely vast

- E.g. to propose new catalyst, need to consider possibilities in
  - Host materials 30 transition metals
  - Dopants 30 transition metals
  - Surface terminations $\simeq 3 - 5$
  - Reaction intermediates $\simeq 10$
  - Adsorption configurations $\simeq 30$

  $\Rightarrow$ $\simeq 10^5 - 10^6$ possibilities!

- State-of-the-art: Density-functional theory (DFT)
- One calculation: $\mathcal{O}(\text{hours})$

So can't I just do machine learning?

- Yes (ongoing research), but ...

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

[2] Z. Ulissi, private communication in ARPAE differentiate group seminar, Dec 2020.

So can't I just do machine learning?

- Yes (ongoing research), but ...
- ...I still need DFT as training data
- Open Catalyst Project[1]
  - 1.3 million DFT calculations
  - > 250 million DFT energy evaluations
  - Workflow success rate: $\simeq 50\%$[2]

$\Rightarrow$ Need high degree of automation

$\Rightarrow$ Reliability needs to be improved!

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

[2] Z. Ulissi, private communication in ARPAE differentiate group seminar, Dec 2020.

So can't I just do machine learning?

- Yes (ongoing research), but ...
- ...I still need DFT as training data
- Open Catalyst Project[1]
  - 1.3 million DFT calculations
  - > 250 million DFT energy evaluations
  - Workflow success rate: $\simeq 50\%$[2]

$\Rightarrow$ Need high degree of automation

$\Rightarrow$ Reliability needs to be improved!

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

[2] Z. Ulissi, private communication in ARPAE differentiate group seminar, Dec 2020.

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

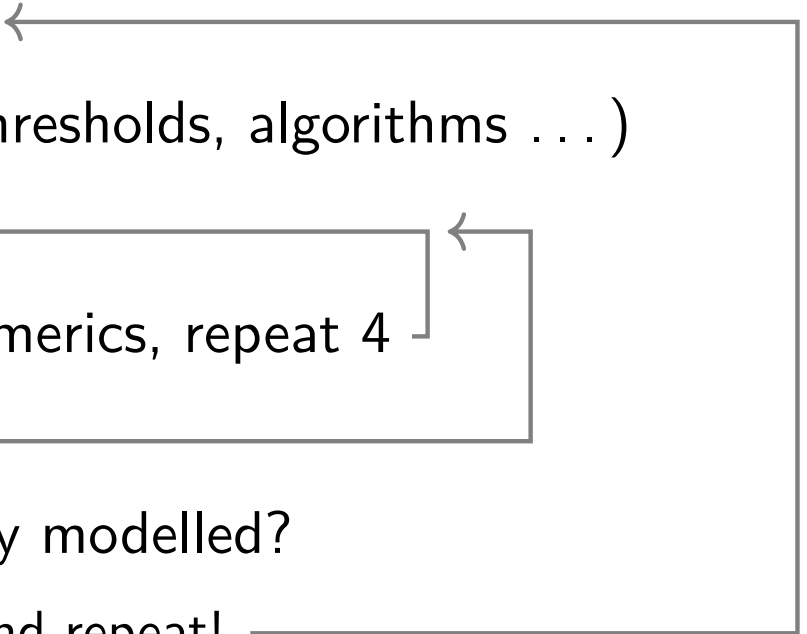
- Euler-Lagrange equations (DFT):

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

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

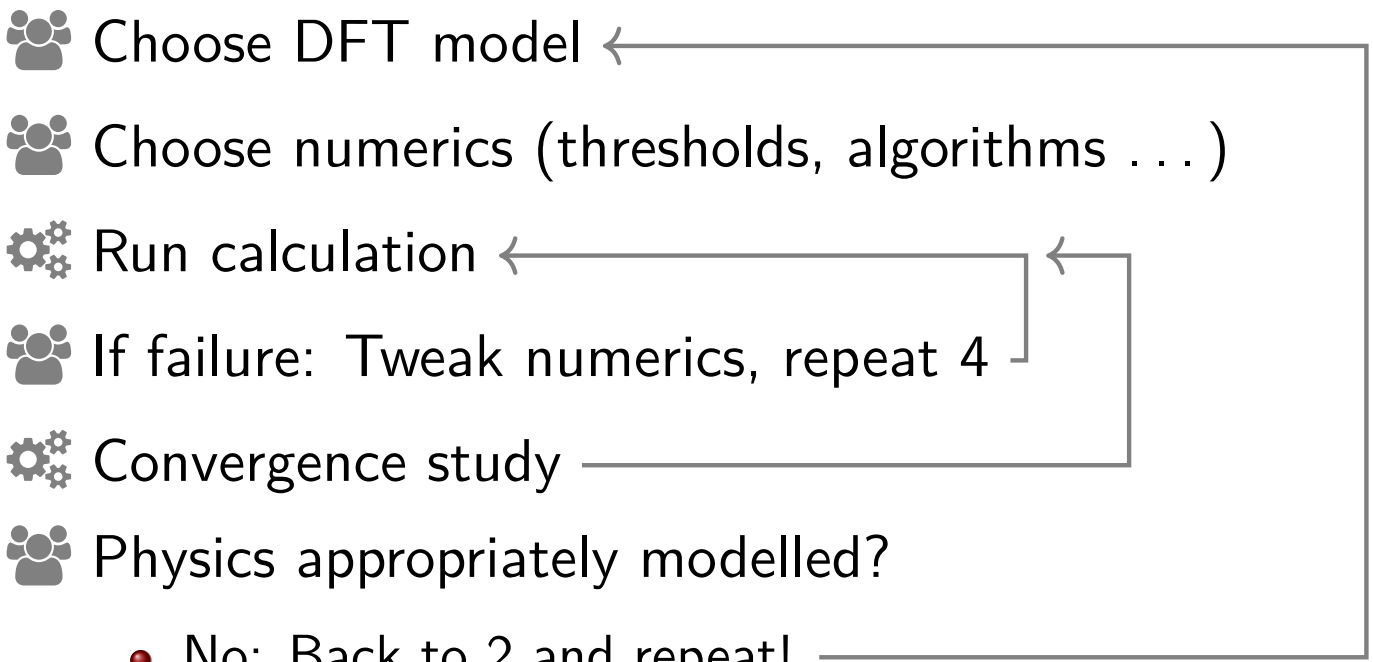
Typical DFT workflow

1. Formulate research question
2. Choose DFT model
3. Choose numerics (thresholds, algorithms . . .)
4. Run calculation
5. If failure: Tweak numerics, repeat 4
6. Convergence study
7. Physics appropriately modelled?
   - No: Back to 2 and repeat!
   - Yes: Hooray! Done!



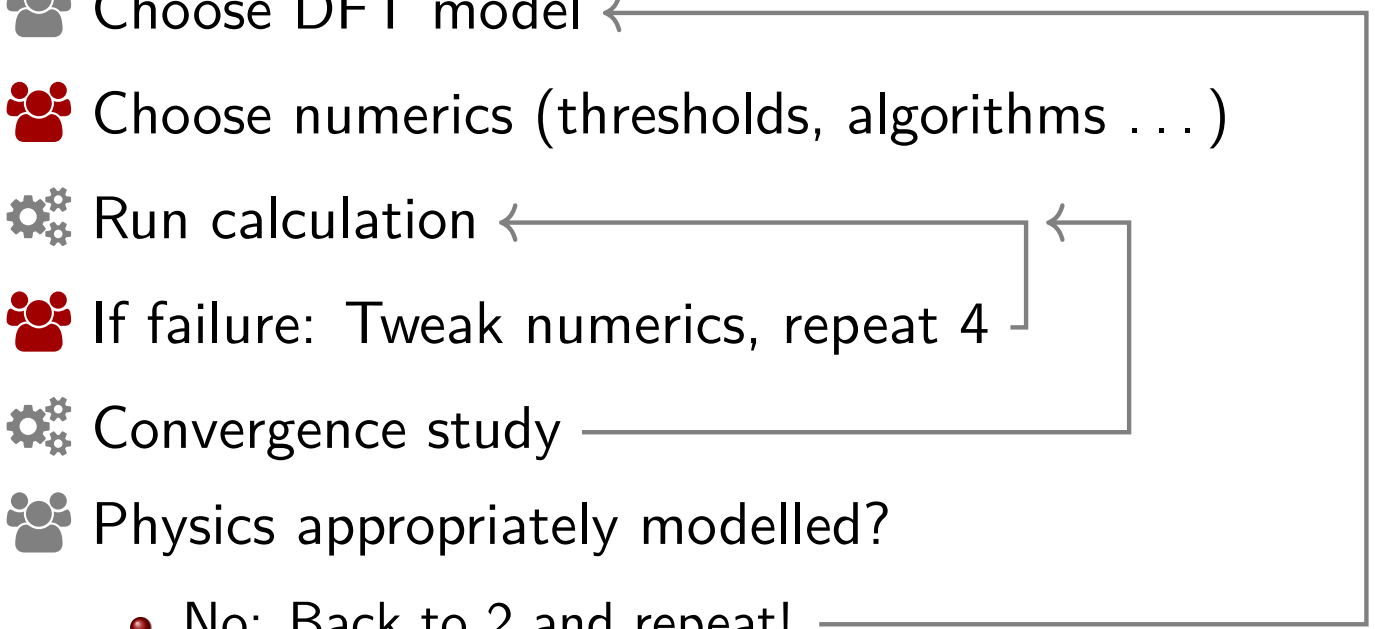
Typical DFT workflow

1. Formulate research question
2. Choose DFT model
3. Choose numerics (thresholds, algorithms . . .)
4. Run calculation
5. If failure: Tweak numerics, repeat 4
6. Convergence study
7. Physics appropriately modelled?
   - No: Back to 2 and repeat!
   - Yes: Hooray! Done!



Typical DFT workflow

1. Formulate research question
2. Choose DFT model
3. Choose numerics (thresholds, algorithms . . .)
4. Run calculation
5. If failure: Tweak numerics, repeat 4
6. Convergence study
7. Physics appropriately modelled?
   - No: Back to 2 and repeat!
   - Yes: Hooray! Done!



Obstacles for high-throughput screening

- 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?

Contents

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 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. *In press.* (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).

A posteriori error analysis in DFT

- *A posteriori* error: Upper bound how far solution is off
- Mathematical answer for: Has target accuracy been reached?
- Sources of error in DFT:
  - Model error
  - Discretisation error
  - Algorithm error
  - Arithmetic error
- *A posteriori* bound on error $\Rightarrow$ Automatic error balancing
- For DFT: Error estimation in its infancy

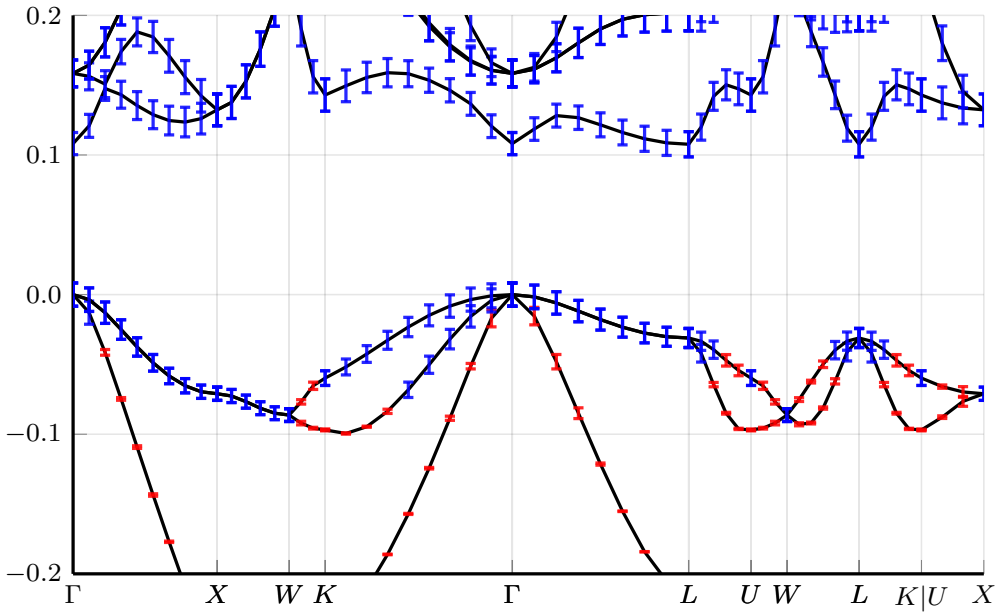
A posteriori error analysis with DFTK

- Requirements:
  - Mathematical theory can only treat reduced models
  - Step-by-step expansion
  - Ingredients not yet clear (e.g. form of integrals, derivatives)

  $\Rightarrow$ Need accessible toolbox for experimentation
  - Arithmetic error: Interval arithmetic, elevated precision
- DFTK offers:
  - Fully customisable model
  - Support for arbitrary floating-point types
  - Use julia ecosystem on DFTK datastructures:
    - Numerical quadrature, forward-mode AD, ...

    $\Rightarrow$ Rapid prototyping in numerical linear algebra

A posteriori error analysis: First results[1]



- Reduced model: Non-self-consistent Kohn-Sham
- Estimation of arithmetic error (`IntervalArithmetic.jl`)
- Used elevated floating-point type (`DoubleFloats.jl`)
- Time to submission: 10 weeks

[1] M. F. Herbst, A. Levitt and E. Cancès. Faraday Discuss. *In press.* (2020)

Reliable SCF algorithms

- Convergence of SCF depends on dielectric properties

⇒ Different SCF needed for metals, insulators, semiconductors

- State-of-the-art schemes based on results for bulk materials

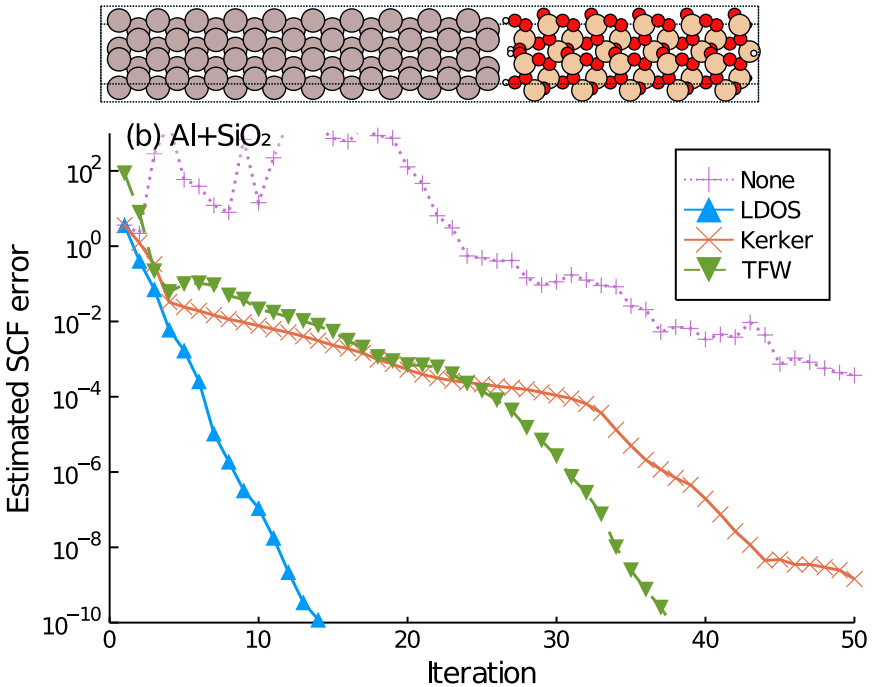
⇒ What to do e.g. for surfaces?

- Requirements:
  - 1D / 2D / 3D: Analyse spectral properties (`KrylovKit.jl`)
  - Rapid prototyping to mix and match ideas

  ⇒ High-level code inside key algorithms
  - Testing requires realistic systems, but first version never works

  ⇒ Scale-up *without* major rewrite of toy code

Black-box preconditioner for inhomogeneous systems[1]



- Black-box and parameter-free preconditioner scheme (LDOS)
- Works in both metallic and insulating regions
- Same implementation for initial exploration and > 800 electrons

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

Lowering the entrance barrier for researchers

- Money: Always tight
- Time: 3-ish years for a PhD, master even less
- State of the art:
  - *Some* codes require software licences $\mathcal{O}(5\text{k€})$
  - Usage: Input format and interface
  - Development: Scarce tests, comments, documentation
  - 1M lines of hardly uniform code conventions
  - Original developers have left (PhD is over)

Attempts to lower the barriers in DFTK

- julia: Zero cost and great learning resources
- Design goal: Code follows mathematical structure of DFT
- Aim for best agreement between code and equations (Unicode)
- Comments: Hint derivation or point to original articles
- `https://docs.dftk.org` with plenty of usage example

- Example projects:
  - Publication following master project
  - 8-week student project to toy with GPUs in DFT

  $\Rightarrow$ Both cases: No familiarity with DFT or julia

  - Error estimates paper

Summary and outlook

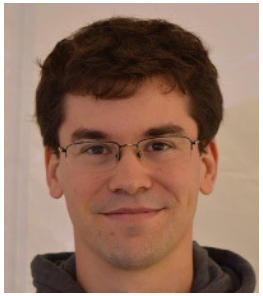
- High-throughput screening
  - Key Ingredients to design the materials of tomorrow
  - Main obstacle: Large number of parameters
  - Chosen empirically ⇒ Reliability limited
- **DFTK**: Multidisciplinary software development
  - Closely integrated with julia ecosystem
  - Mix and match to build new algorithms
  - Toy problems and scale-up to realistic applications
- Future directions . . .

  Reliability Error estimates and black-box workflows

  Efficiency Mixed precision methods

  Hardware usage GPU integration

Acknowledgements



Antoine Levitt

Eric Cancès

Other DFTK contributors:
Eric Berquist, Gaspard Kemlin, Louis Ponet, Sami Siraj-Dine, Zsuzsanna Tóth

Questions?

https://dftk.org

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

mfherbst

https://michael-herbst.com/blog

michael.herbst@inria.fr