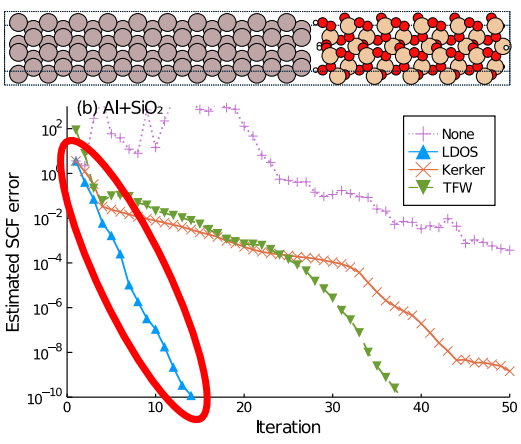
Towards robust error control for high-throughput materials simulations

Michael F. Herbst

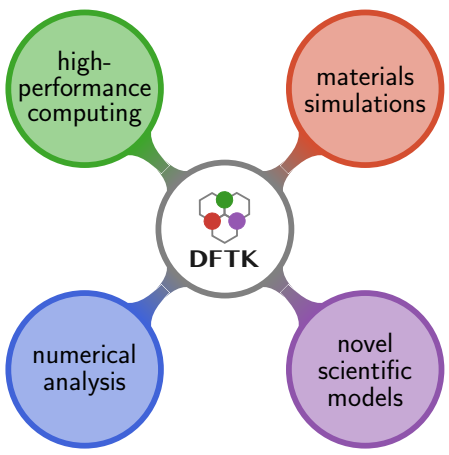
Mathematics for Materials Modelling (`matmat.org`), EPFL

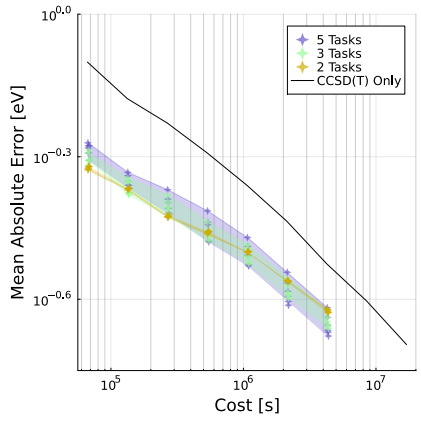
21 November 2023

Slides: `https://michael-herbst.com/talks/2023.11.21_theory_lunch.pdf`



Developing novel DFT algorithms



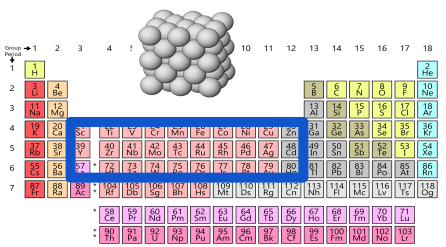


Overcoming model deviations by multi-task learning

Tackling 21st century challenges

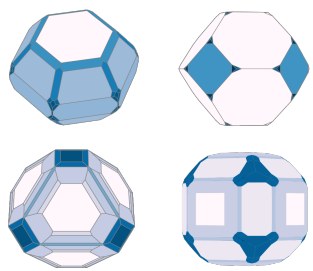
- 21st century challenges:
  - Renewable energy, green chemistry, health care ...
- Current solutions limited by properties of available materials

  $\Rightarrow$ Innovation driven by discovering new materials
- Crucial tool: Computational materials discovery
  - Systematic simulations on $\simeq 10^4 - 10^6$ compounds
  - Complemented by data-driven approaches
  - Noteworthy share of world's supercomputing resources



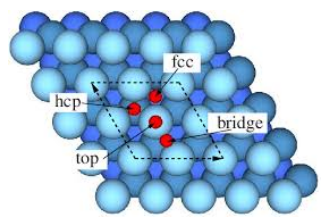
Host metal + dopant

$\simeq 30 \times 30 = 900$



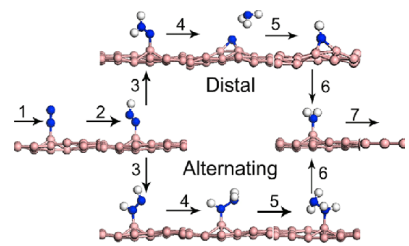
Host surface

$\simeq 3 - 5$



Dopant adsorpotion site

$\simeq 30$



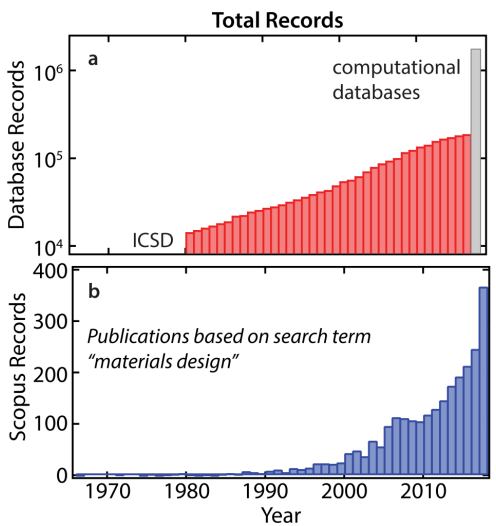
Reaction intermediates

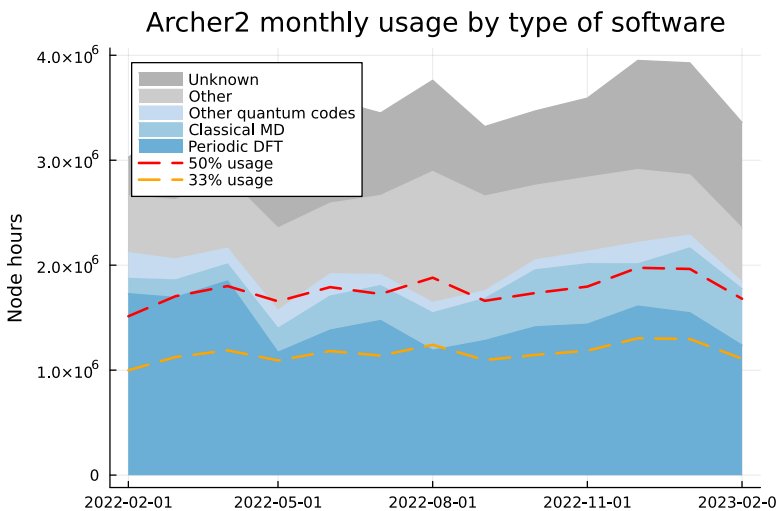
$\simeq 10$

Tackling 21st century challenges

- 21st century challenges:
  - Renewable energy, green chemistry, health care ...
- Current solutions limited by properties of available materials

  $\Rightarrow$ Innovation driven by discovering new materials
- Crucial tool: Computational materials discovery
  - Systematic simulations on $\simeq 10^4 - 10^6$ compounds
  - Complemented by data-driven approaches
  - Noteworthy share of world's supercomputing resources





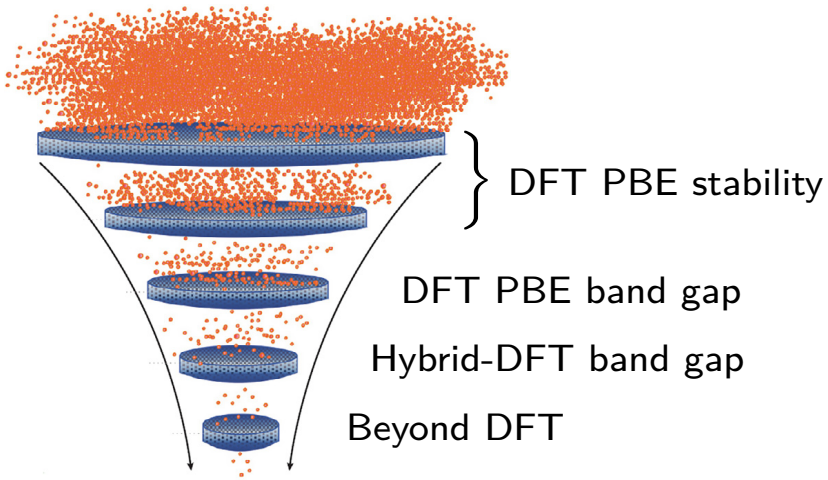
K. Alberi *et. al.* J. Phys. D, **52**, 013001 (2019).

Tackling 21st century challenges

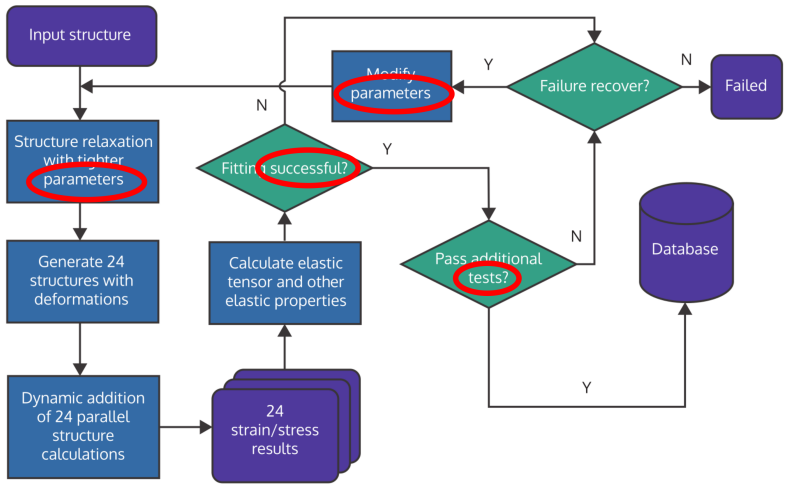
- 21st century challenges:
  - Renewable energy, green chemistry, health care ...
- Current solutions limited by properties of available materials

  $\Rightarrow$ Innovation driven by discovering new materials
- Crucial tool: Computational materials discovery
  - Systematic simulations on $\simeq 10^4 - 10^6$ compounds
  - Complemented by data-driven approaches
  - Noteworthy share of world's supercomputing resources
- Multi-disciplinary effort: Software takes a key role
  - E.g. growing list of data / workflow management tools
  - Challenges of combining efforts & integrating communities


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 throughput)

- **Goal 1** in ~~MMat~~ group: Self-adapting black-box 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).

Broader vision: Robust & error-controlled simulations

- Error control = Track simulation uncertainties:
  - Self-adapting simulations with mathematical guarantees

  $\Rightarrow$ Byproducts: Data quality control, accelerated design

- Error control = Learn missing physics:
  - Data-enhanced models, active learning

  $\Rightarrow$ "Patch up" low-fidelity simulations by select high-fidelity data

- Error control = Leverage inexactness:
  - Error balancing: Optimal adaptive parameter selection
  - Adaptive tolerances & selective precision (16-bit, FPGA)

- **Goal 2** in ~~MatMat~~ group: Estimate and control simulation error
  - Understand where and how to spend efforts best
  - Realm of mathematical research

(Exaggerative) state of codes in this field

Mathematical research

- **Goal:** Numerical experiments
- **Scope:** Reduced models
- High-level **language**:
  Matlab, python, ...
- **Lifetime:** 1 paper
- **Size:** < 1k lines
- Does not care about performance

Application research

- **Goal:** Modelling physics
- **Scope:** All relevant systems
- Mix of **languages:**
  C, FORTRAN, python, ...
- **Lifetime:** 100 manyears
- **Size:** 100k – 1M lines
- Obliged to write performant code

- Working with these codes requires different skillsets
  - $\Rightarrow$ Orthogonal developer & user communities
- Obstacle for knowledge transfer:
  - Mathematical methods never tried in practical setting
    (and may well not work well in the real world)
  - Some issues cannot be studied with mathematical codes
    (and mathematicians may never get to know of them)
- What about emerging hardware, accelerators, performance?
  - Should be the regime of Computer Science (yet another community)

Difficulties of interdisciplinary research

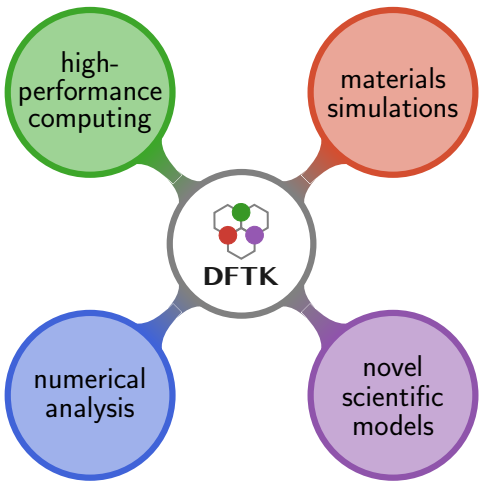
- Community conventions (e.g. publication culture)
- Language barriers and context-sensitive terms
- Speed of research (development of model vs. its analysis)

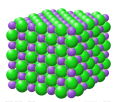
- A social problem . . .
  - (Communication, convention, compromises, . . .)

- . . . that is cemented in software:
  - Priorities differ $\Rightarrow$ What is considered "a good code" differs
  - Insurmountable obstacles to integrate codes
  - Collaborations can stop before they begin . . .

- **Hypothesis:** People compose if software composes

Density-functional toolkit (DFTK) — https://dftk.org





$H\Psi = E\Psi$

- julia-based density-functional theory code
- Cross-community: Mathematical research & applications

- Allows restriction to relevant model problems,
- *and* scale-up to application regime (1000 electrons)

- Integration with multi-scale pipelines:

MARVEL AiiDA & MIT CESMIX

https://nccr-marvel.ch https://cesmix.mit.edu

- Lessons learned:
  - Software integration is hard work
  - Unexpected catalytic effects from integration discussions
  - Each party better understands their role

  $\Rightarrow$ As software composes, communities compose

- **Goal 3** in ~~MatMat~~ group: Lower the barrier for integration
  - Foster cross-community research & fertilisation

Contents

EPFL MATMAT

DFT model classes

- DFT energy minimisation problem:

$$\min_{D\in\mathcal{P}} \mathcal{E}(D) = \min_{D\in\mathcal{P}} \left[\operatorname{tr}(H_0 D) + E_H(\operatorname{diag} D) + E_{\text{xc}}(\operatorname{diag} D)\right]$$

- DFT model hierarchy for E_{xc}: Jacob's ladder
  - LDA, GGA, meta-GGA, Hybrid, RPA-like, Double Hybrid, ...
  - Each *rung* defines (parametrised) model class
- Higher rungs:
  - Generally more expensive, but also more accurate
  - But: DFT is a non-variational approximation to exact physics

  $\Rightarrow$ No guaranteed accuracy order

- **Guiding idea:** Can we combine information from different functionals to balance accuracy / cost / deviating predictions?
- Important: We should not impose an order!

DFT model classes

- DFT energy minimisation problem:

$$\min_{D\in\mathcal{P}} \mathcal{E}(D) = \min_{D\in\mathcal{P}} \left[\operatorname{tr}(H_0 D) + E_H(\operatorname{diag} D) + E_{\mathsf{xc}}(\operatorname{diag} D)\right]$$

- DFT model hierarchy for E_{xc}: Jacob's ladder
  - LDA, GGA, meta-GGA, Hybrid, RPA-like, Double Hybrid, ...
  - Each *rung* defines (parametrised) model class
- Higher rungs:
  - Generally more expensive, but also more accurate
  - But: DFT is a non-variational approximation to exact physics

  $\Rightarrow$ No guaranteed accuracy order
- **Guiding idea:** Can we combine information from different functionals to balance accuracy / cost / deviating predictions?
- Important: We should not impose an order!

Test problem: Ionisation potentials of organic molecules

- Dataset: $\simeq 3000$ small organic molecules[1]
  - ANI-1 subset (2–5 heavy atoms, a few with 6 heavies)
- Targeted quantity: Ionisation potential
  - Note: A challenging quantity for DFT
- Considered models:

	density-functional theory (DFT)			coupled cluster
model	PBE	PBE0	PBE0_DH	CCSD(T)
scaling	$O(N^3)$	$O(N^3)$	$O(N^3)$	$O(N^7)$
advantage	cheap	cheapish	cheapish	accurate
rung	2nd (GGA)	4th (Hybrid)	6th (double Hybrid)	*Reference*

- **Goal:** Surrogate for CCSD(T) but mostly use DFT data

[1] C. Duan, F. Fang, A. Nandy, H. Kulik. J. Chem. Theo. Comput. **16**, 4373 (2020).

Delta learning: Learning to correct the error

- **Idea:** Surrogate for *difference* between high- & low-fidelity
- Gaussian Process (GP) ansatz:

$$I^{\text{CCSD(T)}} - I^{\text{DFT}} = f(\xi) + \varepsilon$$

$$\varepsilon \sim \mathcal{N}(0, \sigma^2 I) \qquad \text{(Gaussian noise)}$$

$$f \sim \mathcal{GP}(\mu, K_\theta) \qquad \text{(GP prior)}$$

ξ: vector of molecular descriptors, I^x: vector of simulated data,
K_θ: Kernel (e.g. polynomial, sq. exponential), σ, μ, θ: hyperparameters

- **Training:** Need DFT & CCSD(T) data
- **Prediction:** Add DFT simulation to predicted mean of GP
- Apply recursively: Multiple levels
- Disadvantages:
  - Ordering imposed
  - All lower levels need to be available

Multitasking: All DFT models are equal

- Asymmetric multitasking[1]:
  - ρ^α: Correlation between CCSD(T) and low-fidelity
  - δ^α: Disparity of low-fidelity models

$$I^\alpha = f^\alpha(\xi) + \varepsilon^\alpha \qquad \text{for model } \alpha \in \{\text{CCSD(T)}, \text{PBE}, \text{PBE0}, \ldots\}$$

$$f^{\text{PBE}}(\xi) = \rho^{\text{PBE}} f^{\text{CCSD(T)}}(\xi) + \delta^{\text{PBE}}(\xi)$$
$$f^{\text{PBE0}}(\xi) = \rho^{\text{PBE0}} f^{\text{CCSD(T)}}(\xi) + \delta^{\text{PBE0}}(\xi)$$

GP prior on f^α & δ^α of different kernel, mean, variance (hyperparams), ε^α is iid noise

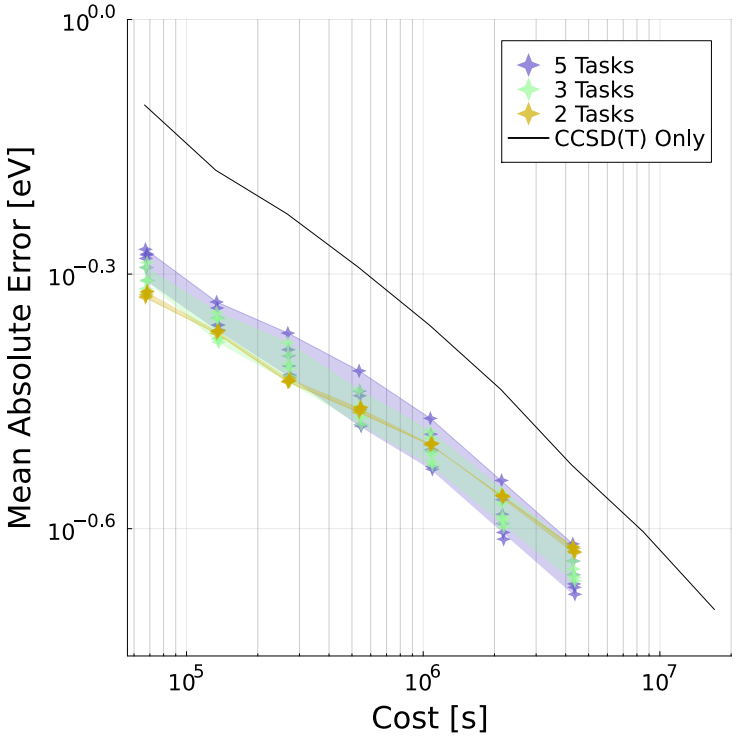
- Rationale:
  - Allow model discrepancies, keep analytical formula for posterior
- Based on small set of calibration data:
  - Fix ρ^α by Pearson correlation
  - Optimise hyperparameters

[1] G. Leen, J. Peltonen, S. Kaski. Mach. Learn. **89**, 157 (2012)

Multitasking: IP results

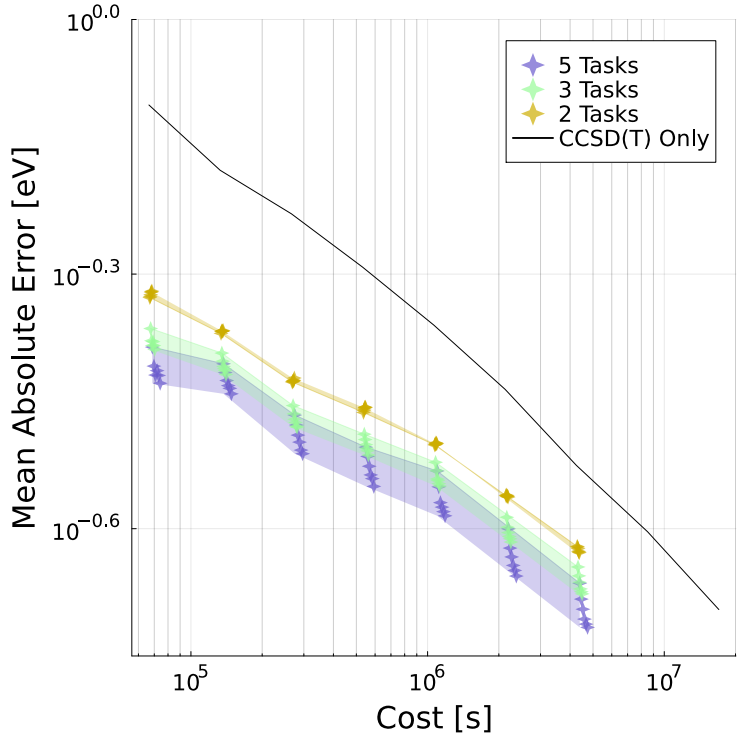
	Core	Additional	Target
CCSD(T)	●●●●●●●●●●		●●●●●●●●●●
PBE0_DH	●●●●	●●●●	
PBE0	●● ●●	●● ●●	
PBE	●● ●●	●● ●●	
BLYP	●● ●●	●● ●●	●●●●●●●●●●

- Goal: Prediction of **T** systems at CCSD(T) level



C and **A** data not shared between tasks

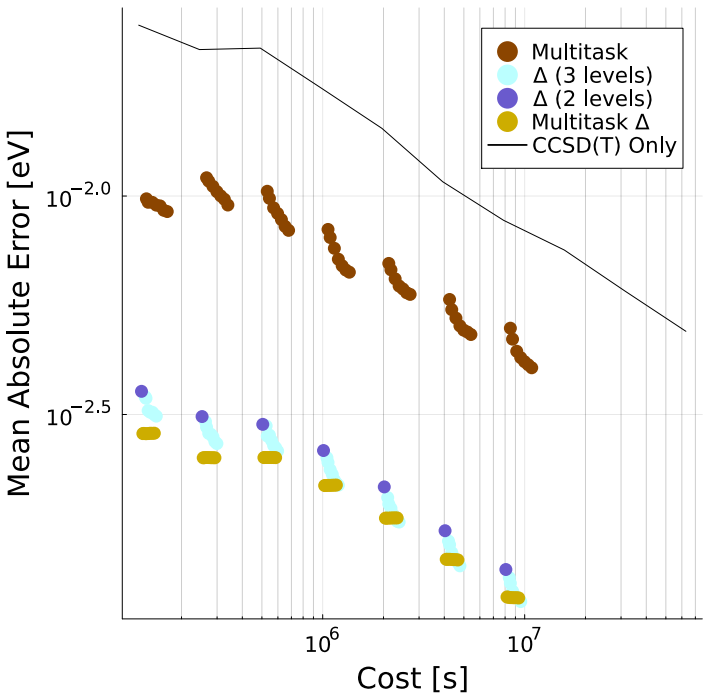
(worst case)



C and **A** data fully shared between tasks

(best case)

Multitasking: Comparison



- Multitask performance depends on correlation between methods
- Different test case (water 3-body energy)
- Δ methods outperforms as difference smoother
- Solution: Multitask-Δ

$\Rightarrow$ Multitask is additional ingredient

Contents

EPFL MATMAT

Density-functional theory (insulators)

- Energy minimisation problem:

$$\min_{D\in\mathcal{P}} \mathcal{E}(D) = \min_{D\in\mathcal{P}} \left[\operatorname{tr}(H_0 D) + E_{\mathsf{Hxc}}(\operatorname{diag} D)\right]$$

with $\mathcal{P} = \left\{D \in \mathfrak{S}_1(L^2) \mid 0 \leq D \leq 1, \operatorname{tr}(D) = N, \operatorname{tr}(-\Delta D) < \infty\right\}$, $[\operatorname{diag} D](\underline{\boldsymbol{r}}) = D(\underline{\boldsymbol{r}}, \underline{\boldsymbol{r}})$

- DFT approximation: Effective single-particle model

$$\begin{cases} \forall i \in 1 \dots N : \left(-\frac{1}{2}\Delta + V(\rho_\Phi)\right)\psi_i = \varepsilon_i \psi_i, \\ V(\rho) = V_{\mathsf{nuc}} + v_C \rho + V_{\mathsf{XC}}(\rho), \\ \rho_\Phi = \sum_{i=1}^{N} |\psi_i|^2, \\ \Phi = (\psi_1, \dots, \psi_N) \in \left(L^2(\mathbb{R}^3, \mathbb{C})\right)^N \text{ orthogonal} \end{cases}$$

nuclear attraction V_{nuc}, exchange-correlation V_{XC}, Hartree potential $-\Delta(v_C\rho) = 4\pi\rho$

$\Rightarrow$ Self-consistent field (SCF) problem: $\rho\big(V(\rho)\big) = \rho$ with

$$\rho(V) = \operatorname{diag}\left[\mathbb{1}_{(-\infty,\varepsilon_F]}\left(-\frac{1}{2}\Delta + V\right)\right] \qquad \text{and } \varepsilon_F \text{ s. t. } \int \rho(V) = N$$

Self-consistent field problem

- Density-mixing SCF procedure (preconditioner P, damping α)

$$\rho_{n+1} = \rho_n + \alpha\, P^{-1} \left[\rho(V(\rho_n)) - \rho_n\right]$$

- Near a fixed-point the error goes as

$$e_{n+1} \simeq \left[1 - \alpha P^{-1}\varepsilon^\dagger\right] e_n$$

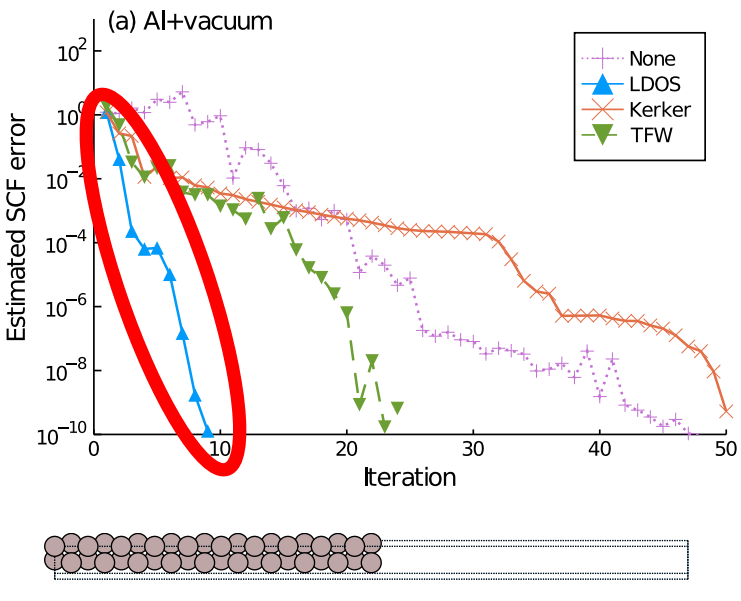
with dielectric matrix $\varepsilon^\dagger = (1 - \chi_0 K)$, $K(\rho) = V'(\rho)$, $\chi_0(V) = \rho'(V)$

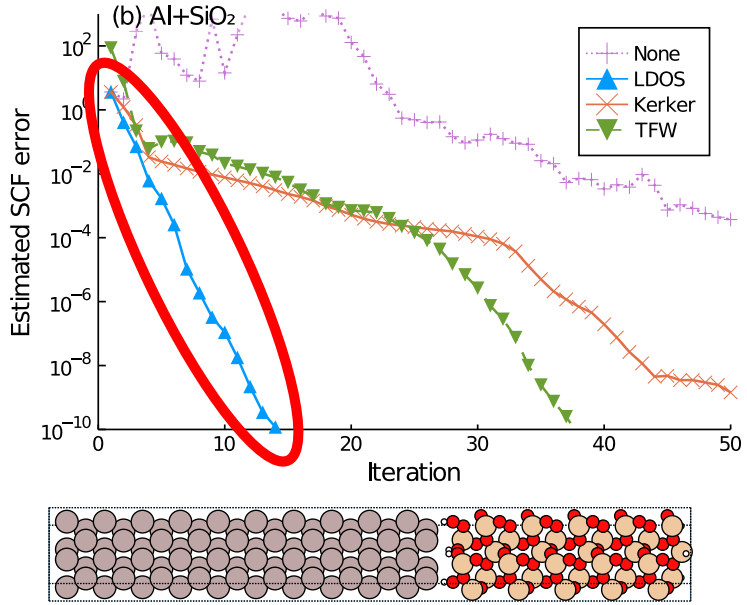
- Convergence iff $-1 < \left[1 - \alpha P^{-1}\varepsilon^\dagger\right] < 1$
  - Dielectric matrix ε: Depends on physics (conduction, screening)
  - By second-order conditions: $\varepsilon \geq 0$ (near fixed point)

$\Rightarrow$ Crucial to design preconditioner such that $P^{-1}\varepsilon \approx I$

  - Note: P need to adapt to physics of unknown system!

LDOS preconditioning (examples)[1]

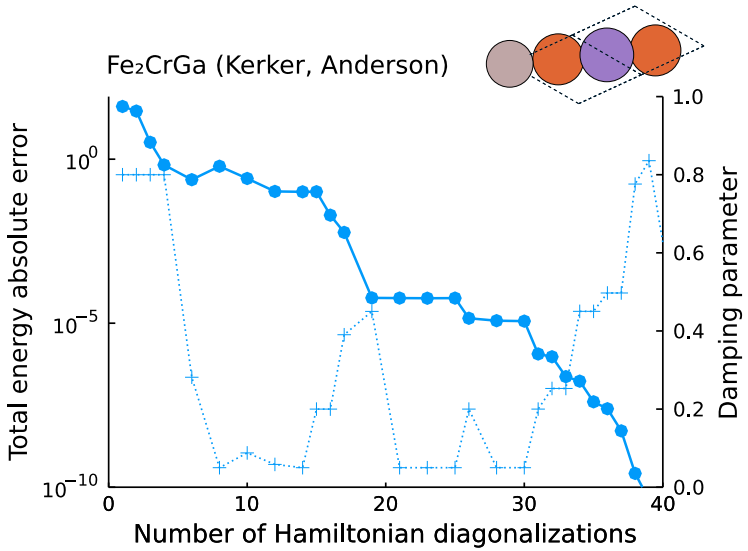




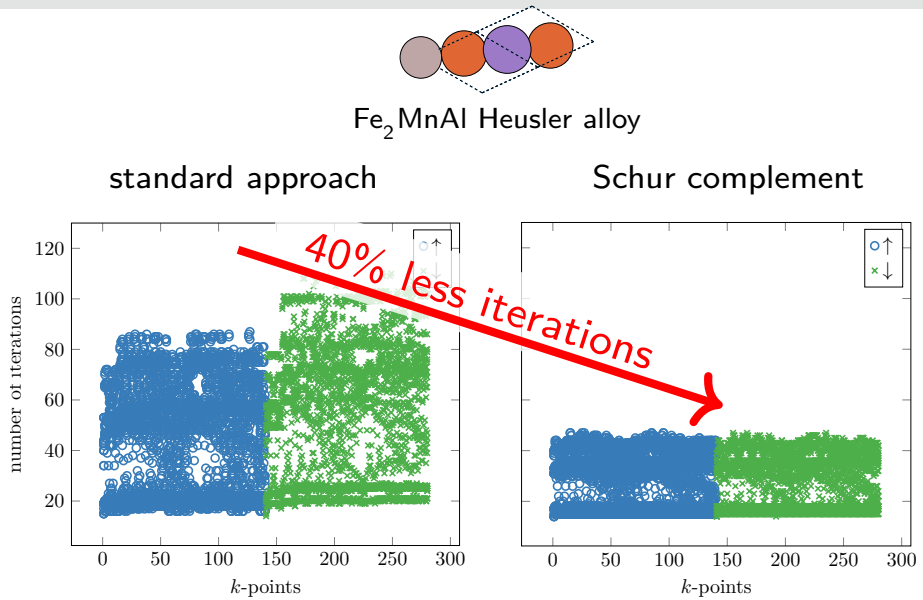
- Inhomogeneous material: Aluminium metal + Insulator
- LDOS automatically interpolates between Kerker mixing (suitable for metals) and no mixing (suitable for insulators)
  - $\Rightarrow$ Based on mathematical understanding of screening
  - $\Rightarrow$ Parameter-free and black-box

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

Robust & efficient algorithms



- Black-box SCF damping α[1]
- α adapted *in each step* using line search & quadratic model
- Novelty: Reuse of expensive quantities in next SCF step
- Reduces trial and error



- First-principle properties of metals
- Schur-complement approach to perturbation theory[2]
  (exploits partially converged states)
- ca. 40% less iterations

⇒ Maths / physics collaboration:
Exchange of ideas between simplified & practical settings crucial

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

[2] E. Cancès, MFH, G. Kemlin, *et. al.* Lett. Math. Phys. **113**, 21 (2023).

DEMO

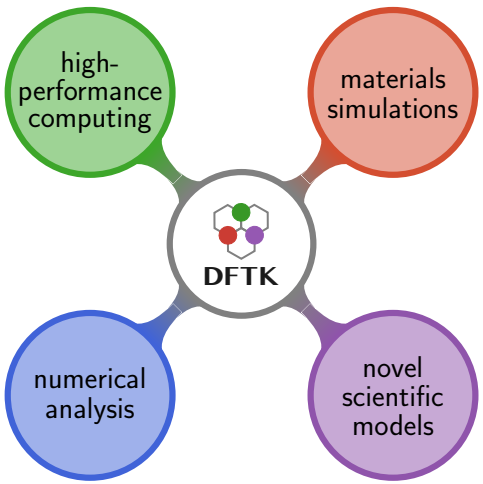
How did  **DFTK** help us to get there?

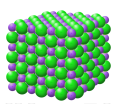
→ `https://michael-herbst.com/talks/2023.11.21_theory_lunch.html`

Contents

EPFL MATMAT

How does DFTK achieve this?





$H\Psi = E\Psi$

- The magic of julia:
  - Separating the what from the how
- Clear design, inspired by mathematical structure
  - $\Rightarrow$ Self-explaining code (a clear what)
- Focus on keeping code accessible (7500 lines)
  - Started in 2018, already 30 contributors
  - Key features by undergrads & outsiders

$\Rightarrow$ High-productivity research framework

$\Rightarrow$ Supports joint research across disciplines

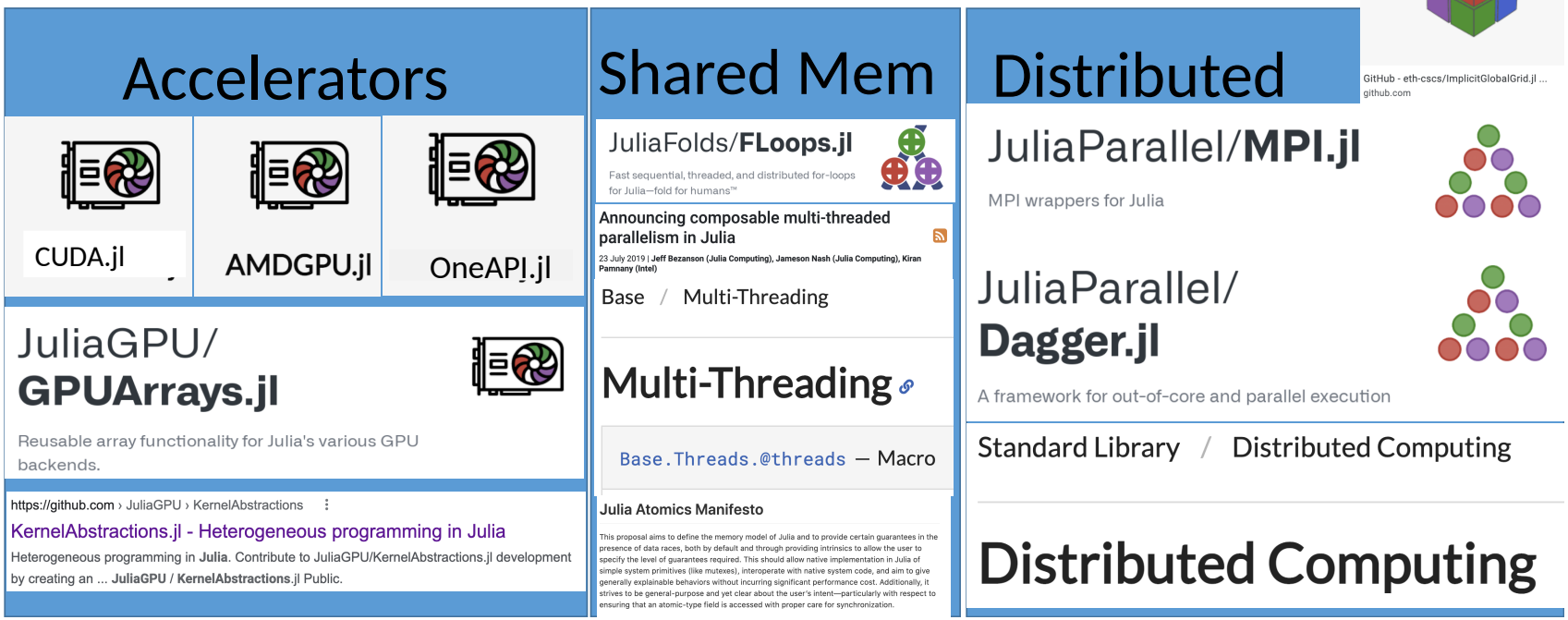
Separating the what from the how

- Why is this separation so important . . .
  - . . . for composable software?
  - . . . for multidisciplinary research?

- Consider the **goal**: Modelling a physical system
- Traditionally users code in detail how the computation should proceed (Imperative programming)
  - How = architecture
  - How = linear algebra primitive (e.g. orthogonalisation)
  - How = memory layout
  - . . .

- But all this has nothing to do with physics!
- Can the how be abstracted away?
  - such that CS / Math can deal with it *independently*
- Let's see julia's HPC developments . . .

julia HPC abstractions




```julia
function power_method(A, x; niter=100)
    for i = 1:niter
        x = A * x
        x ./= norm(x)
    end
    x
end
```

```julia
A = rand(10, 10); A = A + A' + 10I; x = rand(10)

using LinearMaps, IterativeSolvers
itinv(A) = LinearMap(x -> cg(A, x), size(A)...)

using CUDA
power_method(itinv(CuArray(A)), CuArray(x))

using AMDGPU
power_method(itinv(ROCArray(A)), ROCArray(x))
```

DFTK design: Keeping code concise & accessible

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

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

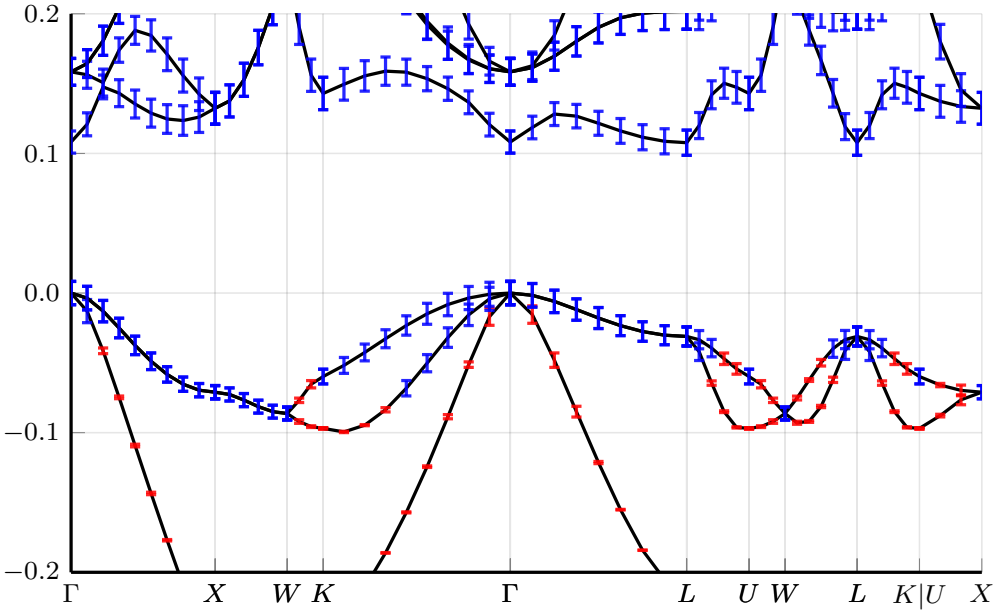
- Stress computation (Definition *vs.* julia code)[1]
- Post-processing step ⇒ Not performance critical
- Comparison of implementation complexity:
  - DFTK: 20 lines[1] (forward-mode algorithmic differentiation)
  - Quantum-Espresso: 1700 lines[2]
  - ≃ 10-week GSoC project

⇒ No performance impact & accessible code

[1] https://github.com/JuliaMolSim/DFTK.jl/blob/master/src/postprocess/stresses.jl

[2] https://github.com/QEF/q-e/blob/develop/PW/src

Support of *a posteriori* error analysis



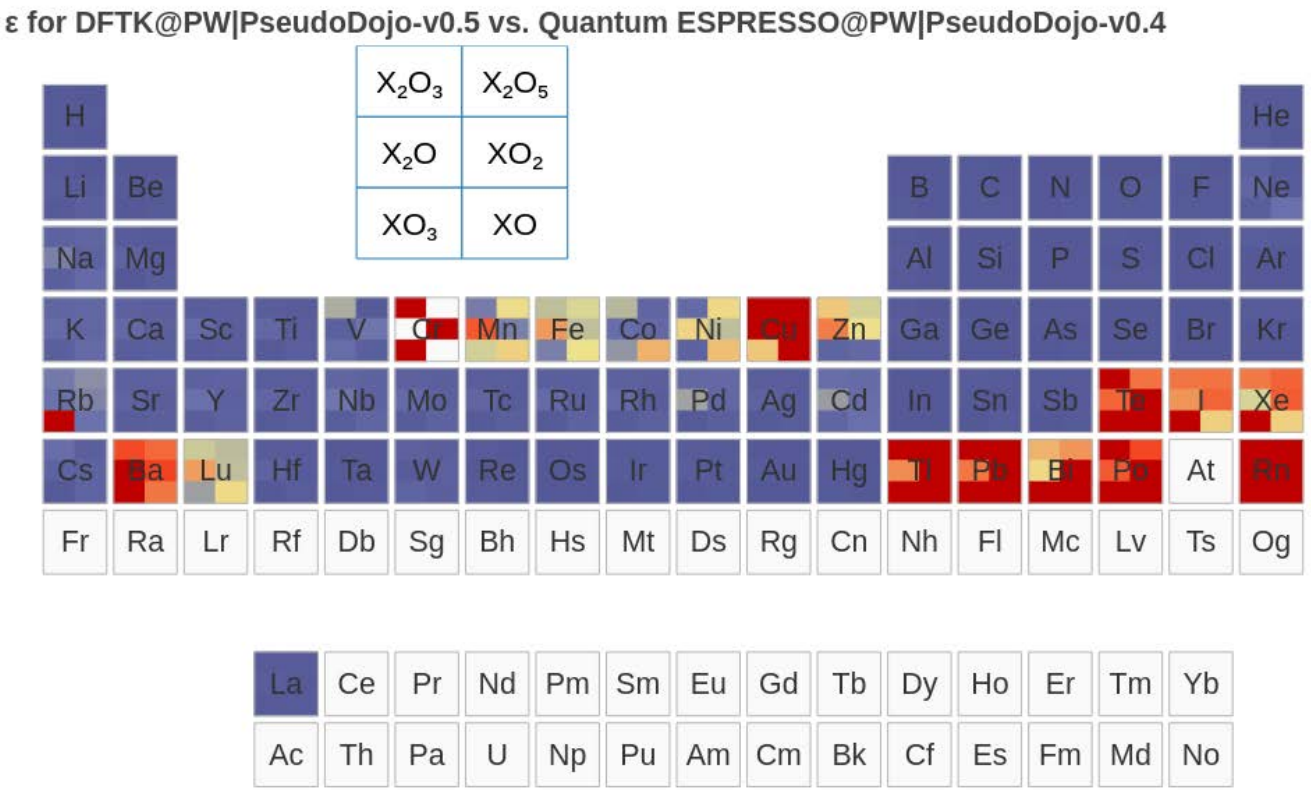
- Albeit the HPC capabilities: Numerical experiments are feasible
- E.g. fully guaranteed error bounds for band structures[1]
- Deals with a reduced Kohn-Sham model and requires interval arithmetic
- Captures basis set error, floating-point error, convergence error
- Recent work based on **DFTK** considers also property errors[2]

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

[2] E. Cancès, G. Dusson, G. Kemlin *et. al.* SIAM J. Sci. Comp., **44**, B1312 (2022).

Integration with AiiDA

- Integration with AiiDA high-throughput workflow manager
  - `https://github.com/aiidaplugins/aiida-dftk`
- Currently used for automated verification tests:



Summary

- Research in the MATMAT group
  - Motivated by high-throughput materials design
  - Need for robust, error-controlled simulation methods
- Multi-tasking surrogate models
  - No need to impose model ordering ⇒ Well-suited for DFT setting
  - Can use cheap model data to compensate for expensive simulations
  - Promising to exploit existing data sets (highly heterogeneous!)
- Black-box strategies for SCF damping & preconditioning
  - Build on combining mathematical and physical insight
  - Safeguard mechanism: Increase robustness for hard cases
- **DFTK**: Multidisciplinary software development
  - julia-based framework for new DFT algorithms
  - In one code: Reduced problems and high-throughput problems
  - High-productivity research framework
  - Overcome disciplinary barriers: People compose if software composes

Acknowledgements

- Niklas Schmitz (MX t Mat)
- Cédric Travelletti (MX t Mat)
- Yihan Wu (EPFL)
- Austin Zadoks (EPFL)
- All DFTK contributors

- Eric Cancès (École des Ponts)
- **Katharine Fisher** (MIT)
- **Antoine Levitt** (Université Paris-Saclay)
- **Youssef Marzouk** (MIT)
- Giovanni Pizzi (PSI)
- Guillaume Vigne (Mines Paris)

EPFL MX t Mat Summer of code

Fonds national suisse

MARVEL NATIONAL CENTRE OF COMPETENCE IN RESEARCH

MIT CESMIX

Questions?

https://matmat.org

mfherbst

michael.herbst@epfl.ch

https://michael-herbst.com/talks/2023.11.21_theory_lunch.pdf

DFTK https://dftk.org

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