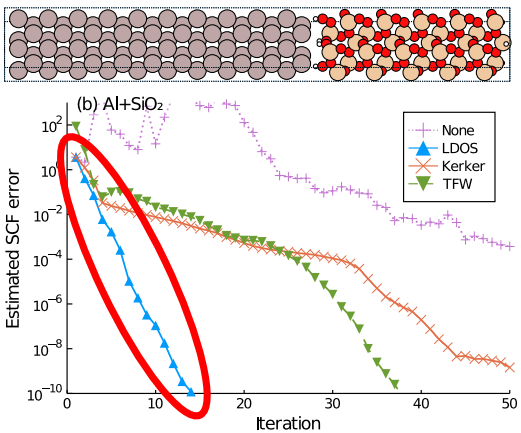
Black-box algorithms and robust error control for density-functional theory

Michael F. Herbst

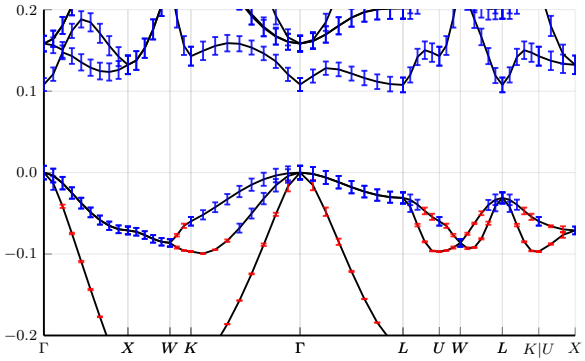
Applied and Computational Mathematics, RWTH Aachen University

18 January 2023

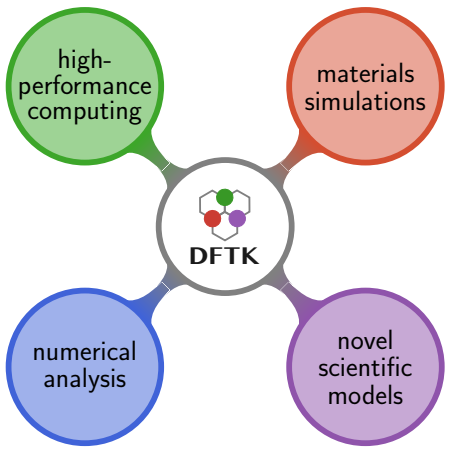
Slides: `https://michael-herbst.com/talks/2023.01.18_error_control_hu.pdf`



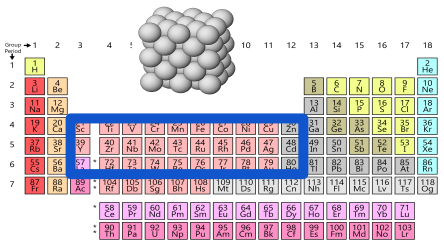
Self-adapting black-box DFT algorithms



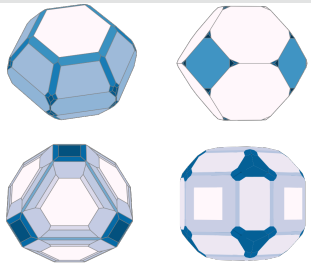
Error control of DFT quantities



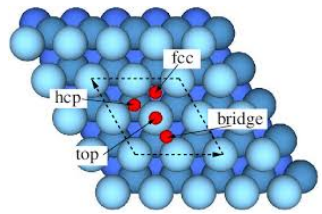
Task: Develop a new metallic catalyst for a surface reaction



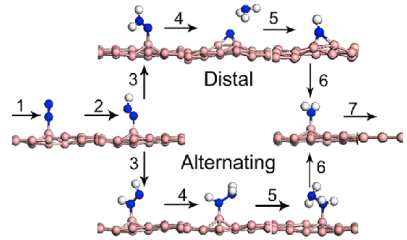
Host metal + dopant
$\simeq 30 \times 30 = 900$



Host surface
$\simeq 3 - 5$



Dopant adsorpotion site
$\simeq 30$



Reaction intermediates
$\simeq 10$

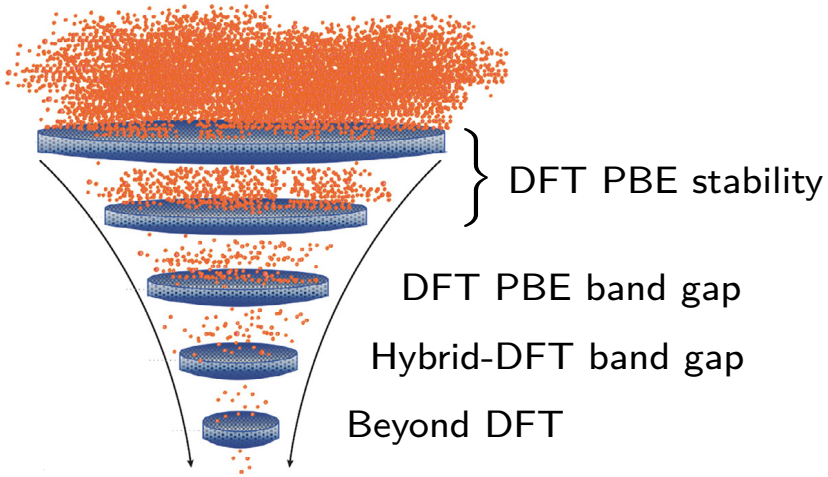
- Combinatorial design space: $\simeq 10^5 - 10^6$ possibilities
- Systematic experiments: Time and cost intensive

$\Rightarrow$ Computational screening to complement and accelerate
- Harvest curated data bases
- Data-driven methods and statistical learning

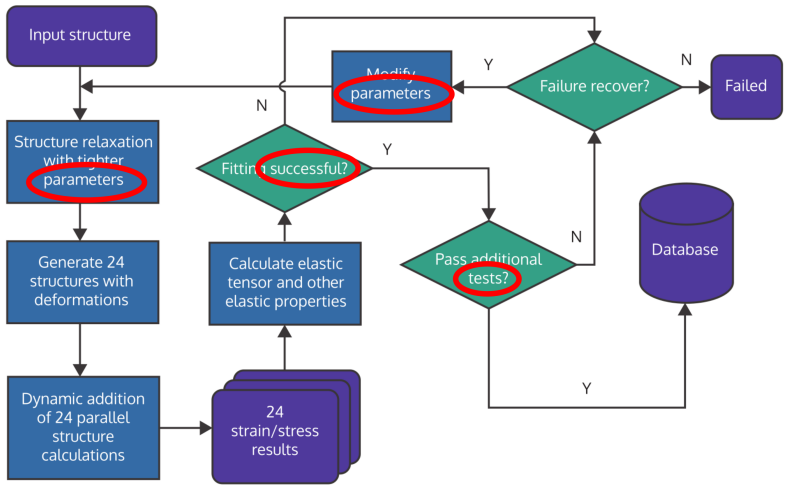
$\Rightarrow$ Regular need of millions of first-principle calculations
- Noteworthy share of world's supercomputing resources
- Growing list of data / workflow management tools

FireWorks AFLOW Automatic-FLOW for Materials Discovery MATERIALSCLOUD AiiDA NOMAD NOVEL MATERIALS DISCOVERY pyiron

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)

- **Goal:** Self-adapting black-box algorithms
  - Transform empirical wisdom to built-in convergence guarantees
  - Requires: Uncertainty quantification & error estimation

  $\Rightarrow$ Understand where and how to spend efforts best

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
  - Integrate with error propagation efforts for surrogates[1]

  $\Rightarrow$ Byproducts: Data quality control, accelerated design
- Error control = Learn missing physics:
  - Data-enhanced models, active learning
  - Integration with experiment (autonomous discovery)

  $\Rightarrow$ Exploit high-fidelity experimental, beyond-DFT data
- Error control = Leverage inexactness:
  - Error balancing: Optimal adaptive parameter selection
  - Randomised methods, selective precision (16-bit, FPGA)
  - Multi-fidelity approaches (reduced basis, surrogates)

$\Rightarrow$ Understand where and how to spend efforts best

$\Rightarrow$ Realm of mathematical research

[1] F. Musil, A. Grisafi *et. al.* J. Chem. Theo. Comput. **15**, 2 (2019).

(Exaggerative) state of codes in this field

Mathematical research

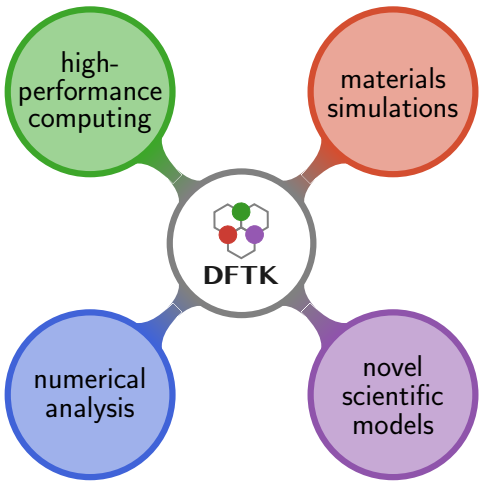
- **Goal:** Numerical experiments
- **Scope:** Reduced models
- High-level **language**:
  Matlab, python, ...
- **Lifetime:** 1 paper
- **Size:** < 1k lines
- Does not care about performance

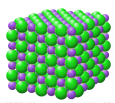
Application research

- **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

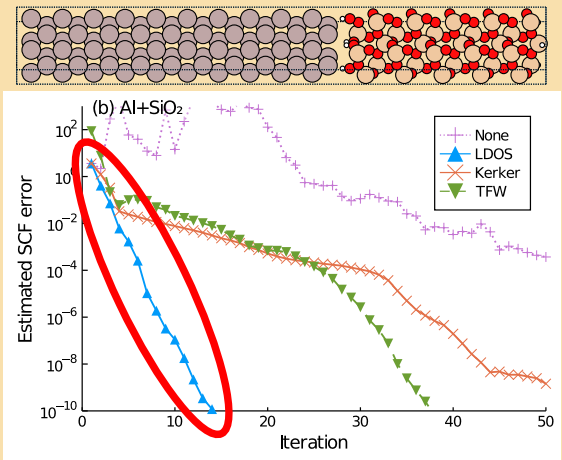




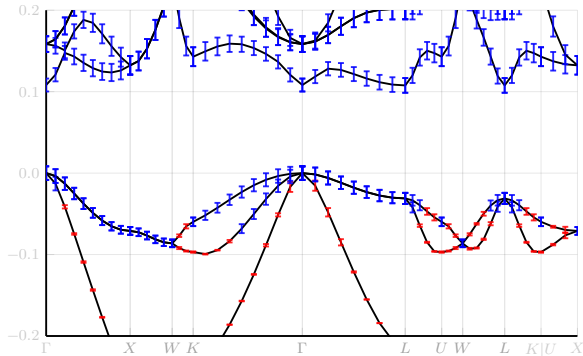
$H\Psi = E\Psi$

- A social problem . . .
  - Community conventions (e.g. publication culture)
  - Language barriers and context-sensitive terms
  - Speed of research (development of model vs. its analysis)
- . . . cemented in software:
  - Priorities differ ⇒ What is considered a "good code" differs
  - Insurmountable obstacles for code integration
  - Collaborations can stop before they begin . . .
- **Hypothesis:** People compose if software composes
- **DFTK**, the Density-Functional ToolKit
  - Allows restriction to relevant model problems,
  - *and* scale-up to application regime (1000 electrons)
  - Sizeable feature set in 7000 lines of code
  - MPI, self-adapting methods, algorithmic differentiation
  - Integrated in multi-scale pipeline
    (potential fitting, molecular dynamics)

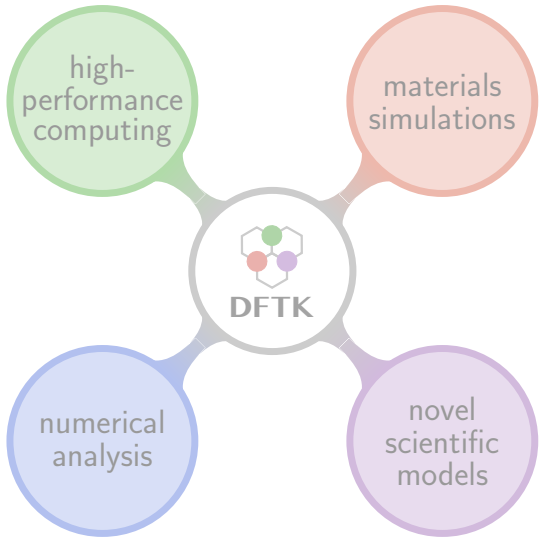
Black-box algorithms and robust error control for DFT



Self-adapting black-box DFT algorithms



Error control of DFT quantities



- Density-functional theory
- Local density of states preconditioner
- Adaptive damping

- Errors in DFT
- Model sensitivitites
- Algorithmic differentiation
- Robust response

- DFTK overview

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: $V\big(\rho(V)\big) = V$ 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

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

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

- In practice: Combined with acceleration (e.g. Anderson)
  - Dropped to simplify analysis
  - Re-introduced for numerical experiments

- Near a fixed-point the error goes as

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

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

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

$\Rightarrow$ Need $P^{-1} \simeq \varepsilon^{-1}$ (matching preconditioner) or small α

Drawback of established approaches

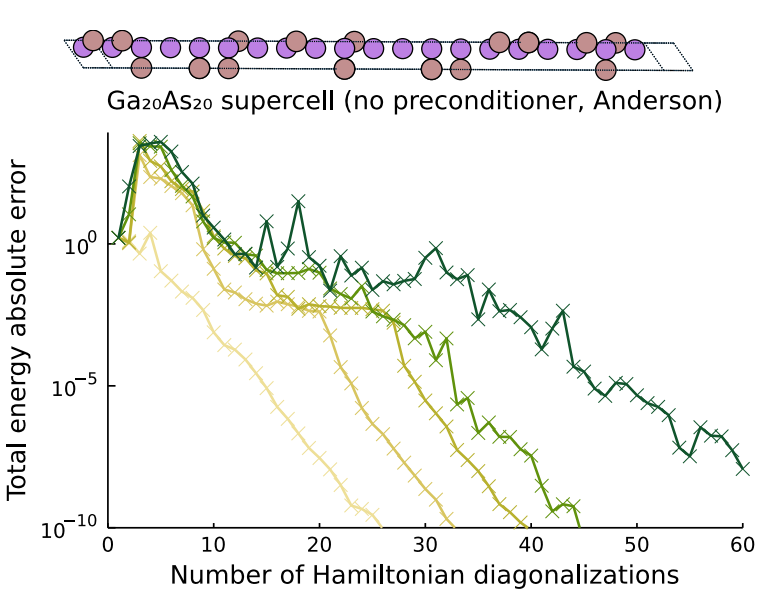
1. Preconditioner P is system-dependent and chosen *a priori*
   - Standard preconditioners: Derived from bulk materials
   - Misses important applications (e.g. inhomogeneous systems)
   - E.g. clusters, passivated surfaces, heterogeneous catalysis, ...

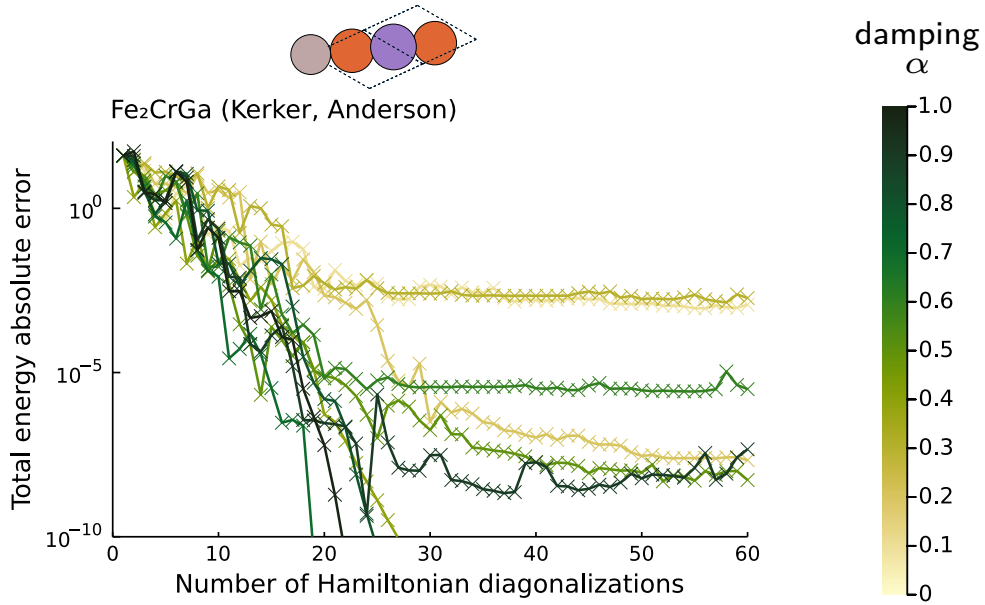
2. If no good preconditioner P known: Trial and error
   - Employ standard heuristics: E.g. **decrease damping** α
   - But: Can fail for interesting cases (the tough 1% ?)

$\Rightarrow$ Wasted computational resources

$\Rightarrow$ **Goal:** Black-box and self-adapting P and α

Illustration: Guessing a suitable damping α can be hard



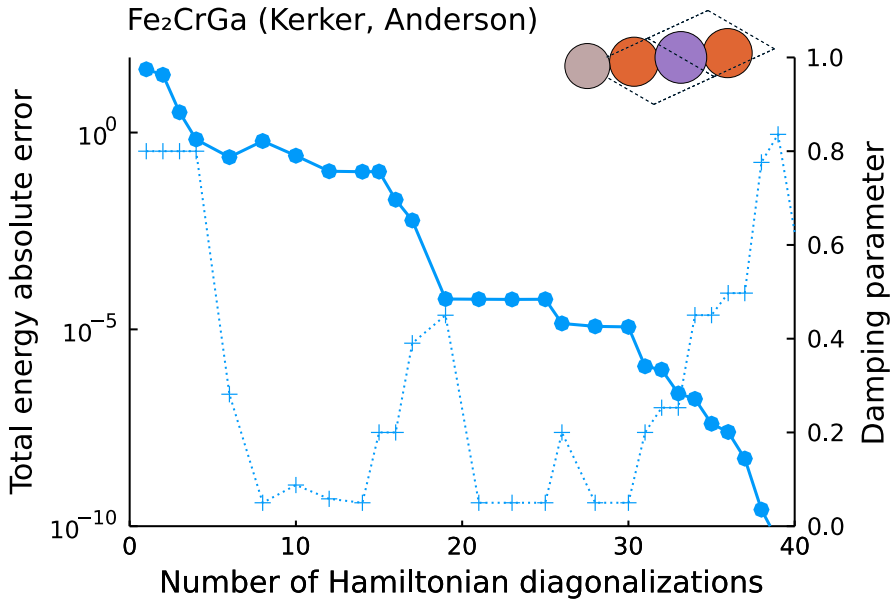


- Inefficient standard damping $(0.6 - 0.8)$
- Surprisingly small damping for smooth convergence

- Heusler alloy: Design space of interest
- Convergence difficulties found in high-throughput studies

- Irregular behaviour: α versus convergence
- Heuristics breaks: Larger damping is better

Black-box α: Adaptive damping[1]

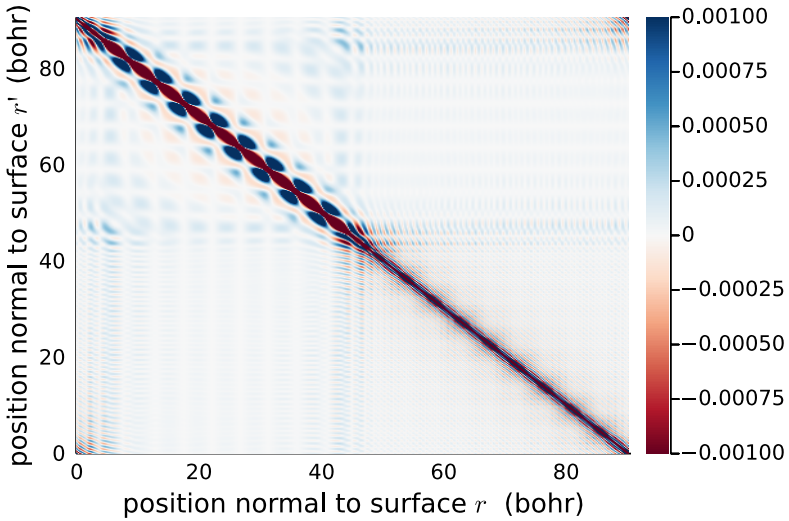


- **Theorem:** SCF convergence guaranteed if α small enough (see paper)
- α adapted *in each step* using line search & quadratic model
- Novelty: Reuse of expensive quantities in next SCF step
  $\Rightarrow$ No overhead if line search immediately successful
- For tricky systems: Adaptive damping has an overhead
  - But: Avoids trial and error
  - Mathematically motivated safeguard mechanism

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

Black-box P: Local density of states (LDOS) mixing[1]

- Bulk preconditioning models approximate inverse $P^{-1} \simeq \varepsilon^{-1}$
- Use $\varepsilon = (1 - K\chi_0)$ with $K(\rho) = V'(\rho), \chi_0(V) = \rho'(V)$
- $\chi_0(r, r')$ unit-cell internal fluctuations, diagonal dominant:



- Tackle charge sloshing: Consider large-scale variations of χ_0:

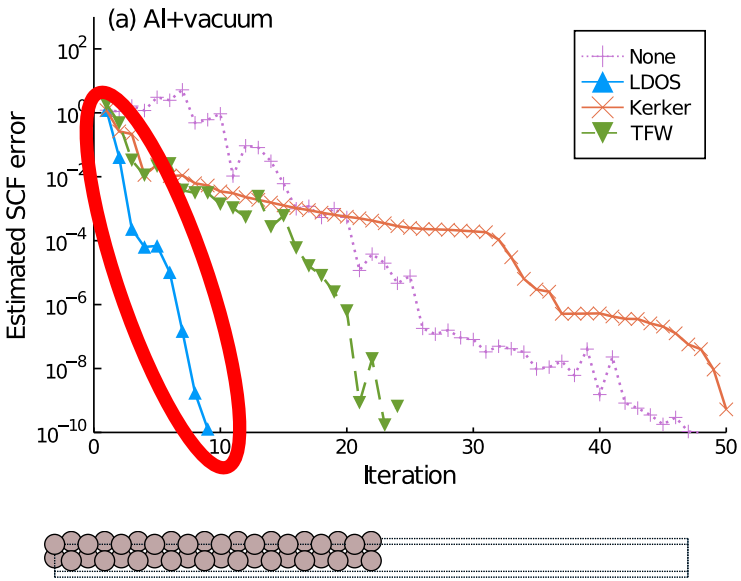
$$\chi_0(r, r') \simeq -\mathrm{LDOS}(r)\delta(r, r') \qquad (\text{homogenisation } \mathrm{LDOS}(r) \approx \int \chi_0(r, r')\,\mathrm{d}r')$$

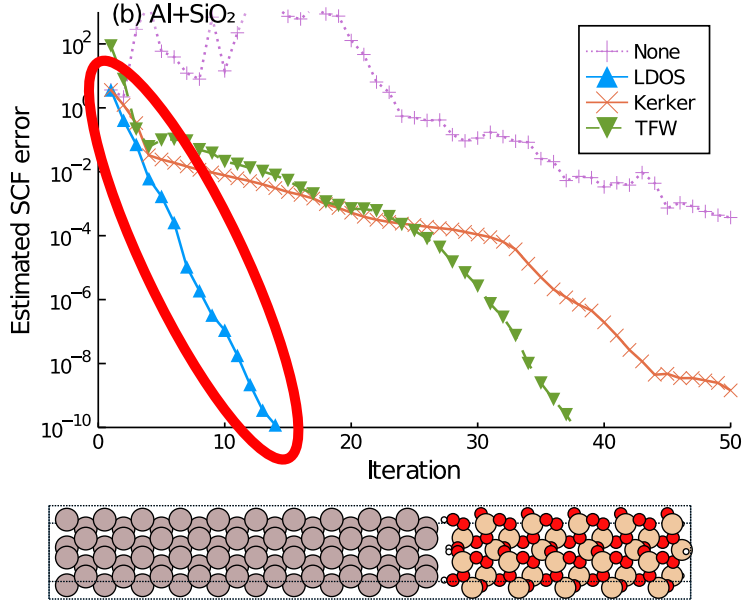
- Apply preconditioner iteratively:

$$P^{-1}V_n = [1 - K\widetilde{\chi_0})]^{-1}\,V_n, \qquad \widetilde{\chi_0}(r, r') = -\mathrm{LDOS}(r)\delta(r, r')$$

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

LDOS preconditioning (examples)[1]



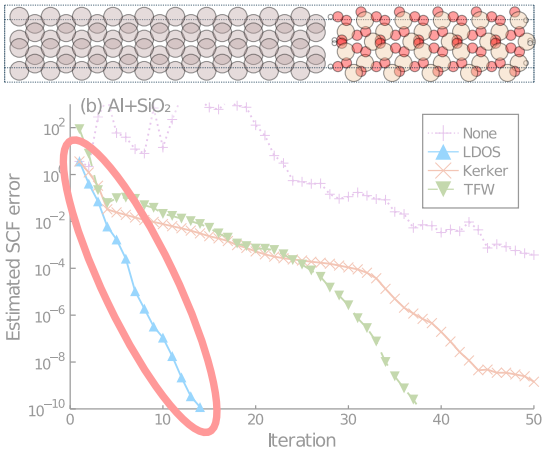


- Inhomogeneous material: Aluminium metal + Insulator
- TFW: local Thomas-Fermi-von Weizsäcker mixing[2]
  (Ad hoc modification of metallic screening model)
- LDOS automatically interpolates between Kerker mixing (suitable for metals) and no mixing (suitable for insulators)
  - ⇒ Based on mathematical understanding of screening
  - ⇒ Parameter-free and black-box

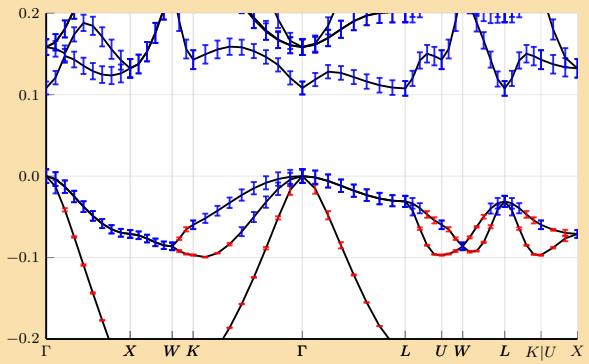
[1] MFH, A. Levitt. J. Phys. Condens. Matter **33**, 085503 (2021).

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

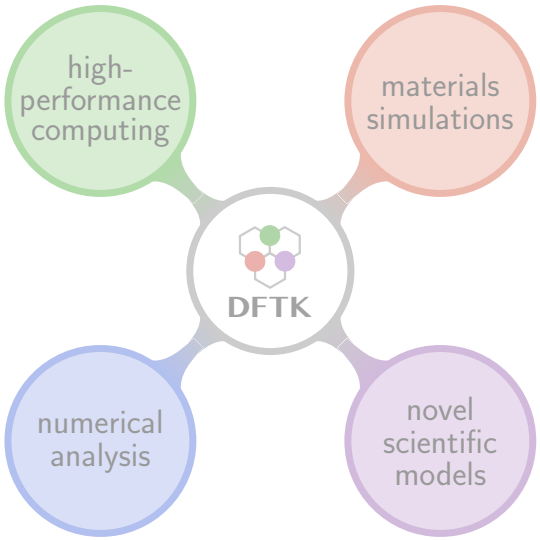
Black-box algorithms and robust error control for DFT



Self-adapting black-box DFT algorithms



Error control of DFT quantities



- Density-functional theory
- Local density of states preconditioner
- Adaptive damping

- Errors in DFT
- Model sensitivitites
- Algorithmic differentiation
- Robust response

- DFTK overview

Error sources in DFT simulations

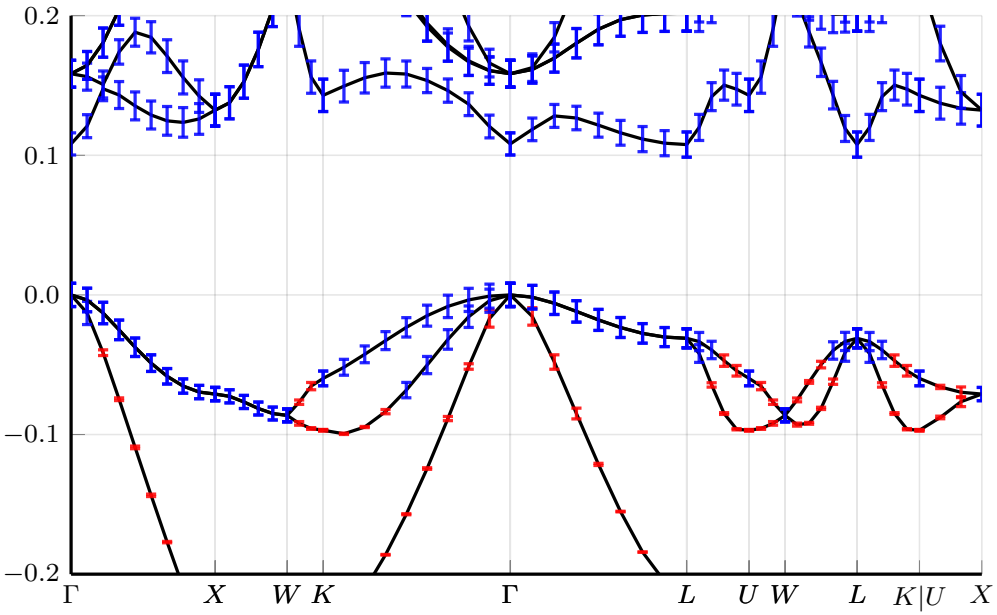
- Model error: Selection of DFT model
- Computational approach:
  - Discretisation error: Basis size, k-point mesh
  - Algorithm error: Convergence thresholds (SCF, eigensolver)
  - Floating-point error: Floating-point arithmetic
- Additionally: Programming error, hardware error (not discussed further)

- Error control: Link parameter selection $\leftrightarrow$ simulation error
  - Enables error balancing
  - Optimised automatic parameter selection

  $\Rightarrow$ Robust, parameter-free & efficient simulations

Status of error control in DFT

- Numerical analysis $\Rightarrow$ Discretisation error
  - Perturbation-based bounds for Gross-Pitaevskii[1] and DFT[2]
  - Current status: Mostly restricted setting & simplified models
  - Guaranteed bounds for band structures in a pseudopotential model[3]
  - Captures basis set error, floating-point error, convergence error



[1] E. Cancès, G. Dusson *et. al.* Comp. Rend. Math. **352**, 941 (2014).

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

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

Status of error control in DFT

- Numerical analysis ⇒ Discretisation error
  - Perturbation-based bounds for Gross-Pitaevskii[1] and DFT[2]
  - Current status: Mostly restricted setting & simplified models
  - Guaranteed bounds for band structures in a pseudopotential model[3]
  - Captures basis set error, floating-point error, convergence error
- Statistical techniques ⇒ Model error
  - Ensemble-mediated (BEEF)[4]
  - Representative comparison (Δ-test)[5]
  - Focus here: Model sensitivities by algorithmic differentiation

[1] E. Cancès, G. Dusson *et. al.* Comp. Rend. Math. **352**, 941 (2014).

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

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

[4] V. Petzold, T. Bligaard *et. al.* Top. in Catal., **55**, 402 (2012).

[5] K. Lejaeghere, G. Bihlmayer *et. al.* Science, **351**, aad3000 (2016).

DFT response properties and their sensitivities

- Illustration: DFT Hamiltonian $H_{a\theta}$
  - a: Lattice constant
  - θ: DFT exchange-correlation parameters
- Self-consistent field yields fixed-point density ρ_{SCF}

$$0 = \mathrm{diag}\left[\mathbb{1}_{(-\infty,\varepsilon_F]}(H_{a\theta}(\rho_{\mathrm{SCF}}))\right] - \rho_{\mathrm{SCF}}$$

- Defines implicit function $\rho_{\mathrm{SCF}}(a,\theta)$
- DFT properties: Response of system to external perturbation

  $\Rightarrow$ (Higher-order) derivative of some function of ρ_{SCF}
  - Examples: Forces (energy wrt. position), dipole moment (energy wrt. el. field), elasticity (energy cross-response to lattice deformation)
  - To get sensitivities: Unusual & higher derivative orders needed (i.e. not available in standard codes)

Computing sensitivities

- Consider model sensitivity of stress $S(a,\theta) = \frac{\partial \mathcal{E}(\rho_{\mathrm{SCF}}(a,\theta))}{\partial a}$:

$$\frac{dS}{d\theta} = \frac{\partial S}{\partial \rho_{\mathrm{SCF}}} \frac{\partial \rho_{\mathrm{SCF}}}{\partial \theta} \tag{1}$$

- Computed by implicit differentiation (Dyson equation):

$$\frac{\partial \rho_{\mathrm{SCF}}}{\partial \theta} = [1 - \chi_0 K]^{-1} \chi_0 \frac{\partial H_{a\theta}}{\partial \theta}$$

- Parameters appear in innermost layer (model definition)
  - Each DFT model: Different derivatives $\frac{\partial H_{a\theta}}{\partial \theta}$ (can be horrible)
  - Each quantity of interest: Different sensitivity expression (1)

  $\Rightarrow$ Combinatorial explosion

- Use algorithmic differentiation (AD) ($\approx$ automatic derivatives)
  - Generic framework for DFT derivatives / response properties
  - Saves manual coding: Request gradient (1), AD delivers

  $\Rightarrow$ Breaks "one PhD student per derivative" paradigm

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

How does algorithmic differentiation (AD) work?

- $F : \mathbb{R}^2 \to \mathbb{R}$ with
  $F(x) = \text{double}(\text{sum}(x_1, x_2))$

```
function F(x)
    y1 = x[1] + x[2]   # F1 = sum
    y2 = 2 * p         # F2 = double
    return y2
end
```

- Derivative at $\tilde{x}$:

$$[J_F(\tilde{x})]_{ij} = \left(\frac{\partial F}{\partial x}\bigg|_{x=\tilde{x}} \right)_{ij} = \frac{\partial F_i}{\partial x_j}\bigg|_{x=\tilde{x}}$$

- AD: Compose Jacobian of F from known Jacobians of primitives of computational graph:

$$\begin{aligned} e_i^T J_F e_j &= e_i^T \left(J_{\text{double}} J_{\text{sum}} e_j \right) && \text{“Forward mode”} \\ &= \left(J_{\text{sum}}^T J_{\text{double}}^T e_i \right)^T e_j && \text{“Adjoint mode”} \end{aligned}$$

- Devil is in the details:
  - E.g. adjoint mode: Reverse control flow (“back propagation”)

Forward-mode algorithmic differentiation

```
function F(x)
    y1 = x[1] + x[2]   # F1 = sum
    y2 = 2 * p         # F2 = double
    return y2
end
```

$$F(x) = \mathsf{double}(\mathsf{sum}(x_1, x_2))$$
$$e_i^T J_F e_j = e_i^T J_{\mathsf{double}} J_{\mathsf{sum}} e_j$$

- Forward-diff: Evaluate in order with *primal* F:
  1. Set $y_0 = (x_1, x_2)$, $\dot{y}_0 = e_j$
  2. Compute $y_1 = \mathsf{sum}(y_0)$ and $\dot{y}_1 = J_{\mathsf{sum}}(y_0)\dot{y}_0$
  3. Compute $y_2 = \mathsf{double}(y_1)$ and $\dot{y}_2 = J_{\mathsf{double}}(y_1)\dot{y}_1$
  4. Obtain $F(x_1, x_2)$ as y_2 and $[J_F]_{:,j} = \dot{y}_2$

$\Rightarrow$ Obtain one column of J_F at a time

- Advantage: Numbers $\rightarrow$ dual numbers (easy if generic code)
- Disadvantage: Gradients require $\mathcal{O}(N)$ times primal cost

Adjoint-mode algorithmic differentiation

```
function F(x)
    y1 = x[1] + x[2]   # F1 = sum
    y2 = 2 * p         # F2 = double
    return y2
end
```

$$F(x) = \text{double}(\text{sum}(x_1, x_2))$$

$$e_i^T J_F e_j = \left(J_{\text{sum}}^T J_{\text{double}}^T e_i \right)^T e_j$$

- Adjoint-mode AD: Derivative in reverse instruction order.
- *Forward pass*:
  1. Set $y_0 = (x_1, x_2)$
  2. Compute $y_1 = \text{sum}(y_0)$ and store it
  3. Compute $y_2 = \text{double}(y_1)$ and store it
- *Reverse pass*:
  1. Set $\bar{y}_2 = e_i$
  2. Compute $\bar{y}_1 = [J_{\text{double}}(y_1)]^T \bar{y}_2$
  3. Compute $\bar{y}_0 = [J_{\text{sum}}(y_0)]^T \bar{y}_1$
- Obtain $[J_F]_{i,:}$ as $\bar{y}_0^T \implies$ One row at a time, $\mathcal{O}(1)$ times primal

Lattice constant sensitivities in DFTK

```
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])
```

(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

- Optimal lattice constant sensitivities in one line of code

$$a_* = \arg\min_a \mathcal{E}(a, \theta) \qquad \text{sensitivities} = \frac{da_*}{d\theta}$$

- Practical challenges for derivation and implementation:
  - Nested iterative methods (eigensolver, SCF, lattice optimisation)
  - Unusual second-order derivatives (e.g. $\frac{\partial S}{\partial \theta} = \frac{\partial^2 \mathcal{E}}{\partial \theta \partial a}$)
  - Support for future DFT models? (with their different parameters θ)
- DFTK key achievements:
  - Integration with julia's composable AD frameworks
  - Floating-point agnostic design
  - Stable & generic response solver (including metals etc.)
- Fully flexible in DFT model or targeted quantity:
  - User just codes XC energy expression & SCF postprocessing code

Solving response equations

$$\frac{\partial \rho_{\mathrm{SCF}}}{\partial \theta} = [1 - \chi_0 K]^{-1} \chi_0 \frac{\partial H_{a\theta}}{\partial \theta} \tag{2}$$

- Solving (2) (Dyson equation) is not cheap
- Each *application* of χ_0 to a δV requires iteratively solving (3) (Sternheimer equation) for all $n = 1, \ldots, N$

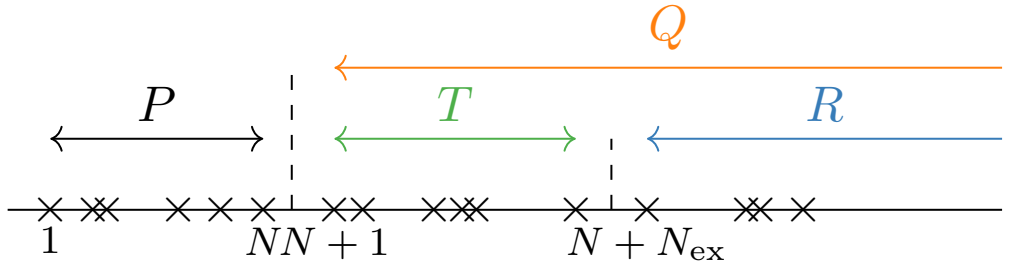
$$\Pi_Q (H - \varepsilon_n) \Pi_Q \delta\psi_n = \underbrace{-\Pi_Q \delta V \psi_n}_{:= b_n} \tag{3}$$

  where
  - $\delta\psi_n$: Orbital perturbation (to be determined)
  - $P = \mathrm{span}\{\psi_n \,|\, n = 1, \ldots, N\}$: Space spanned by N lowest eigenpairs (ε_n, ψ_n) of H (occupied subspace)
  - $\Pi_Q = 1 - \Pi_P$ with Π_P projector onto P.

- Caveat: (3) is badly conditioned if gap $\varepsilon_{N+1} - \varepsilon_N$ small

  $\Rightarrow$ Response can be expensive for metals

Schur-complement approach[1] (1)



- SCF diagonalisations yield N_{ex} additional orbitals $\Phi = (\psi_{N+1}, \ldots, \psi_{N+N_{\text{ex}}})$ spanning T.
  - Not fully converged, i.e. $H\psi_n \neq \varepsilon_n \psi_n$ for $n = N+1, \ldots, N+N_{\text{ex}}$
  - But: $\Phi^T H \Phi = \text{diag}(\varepsilon_{N+1}, \ldots, \varepsilon_{N+N_{\text{ex}}})$
- Split orbital perturbation $\Pi_Q \delta\psi_n = \Phi\alpha_n + \Pi_R \delta\psi_n^R$ to obtain:

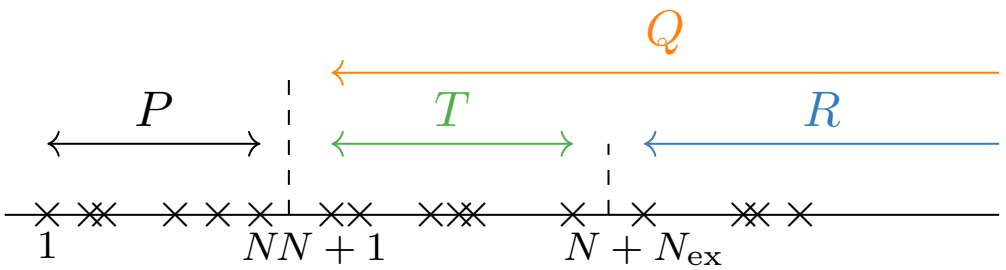
$$\Pi_Q (H - \varepsilon_n)\Phi\alpha_n + \Pi_Q (H - \varepsilon_n)\Pi_R \delta\psi_n^R = b_n$$

- Schur complement: Solve component in T (along Φ) explicitly:

$$\alpha_n = \underbrace{\left(\Phi^T H \Phi\right)^{-1}}_{=D^{-1}} \left(\Phi^T b_n - \underbrace{\Phi^T \left(H - \varepsilon_n\right) \Pi_R}_{=h_{RT}^T} \delta\psi_n^R\right)$$

[1] E. Cancès, MFH, *et. al. Num. stability and efficiency of response property calculations in DFT* arXiv:2210.04512.

Schur-complement approach[1] (2)



$$\Pi_Q(H-\varepsilon_n)\Phi\alpha_n + \Pi_Q(H-\varepsilon_n)\Pi_R\delta\psi_n^R = b_n$$

$$\alpha_n = D^{-1}\left(\Phi^T b_n - h_{RT}^T\delta\psi_n^R\right)$$

- Insert α_n back and project with Π_R from the left:

$$\Pi_R(H-\varepsilon_n)\Phi\left[D^{-1}\left(\Phi^T b_n - h_{RT}^T\delta\psi_n^R\right)\right] + \Pi_R(H-\varepsilon_n)\Pi_R\delta\psi_n^R = \Pi_R b_n$$

$$\Rightarrow \left[\Pi_R(H-\varepsilon_n)\Pi_R - h_{RT}D^{-1}h_{RT}^T\right]\Pi_R\delta\psi_n^R = \left[\Pi_R - h_{RT}D^{-1}\Phi^T\right]b_n$$

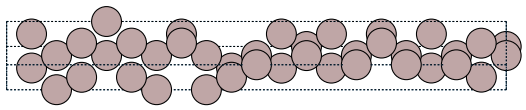
- This can be solved for $\delta\psi_n^R$ using CG
- Φ are almost eigenvectors of H

$\Rightarrow$ Π_R almost removes small eigenmodes of $H-\varepsilon_N$

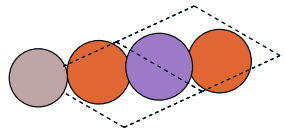
$\Rightarrow$ Improved conditioning

[1] E. Cancès, MFH, *et. al. Num. stability and efficiency of response property calculations in DFT* arXiv:2210.04512.

Schur-based response: Numerical examples[1]

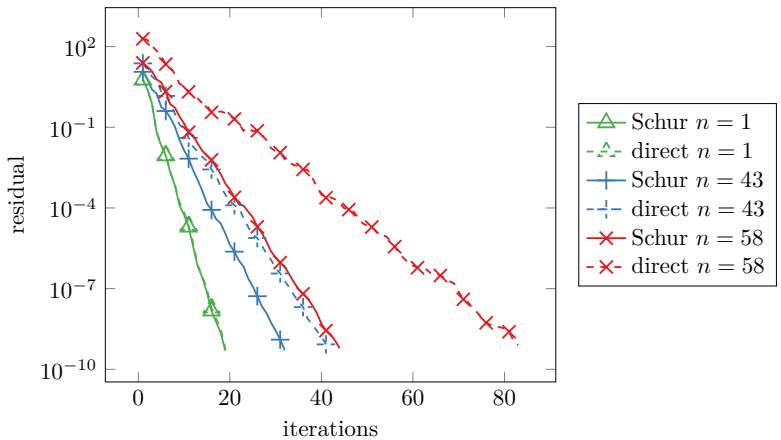


Al_{40} rattled supercell

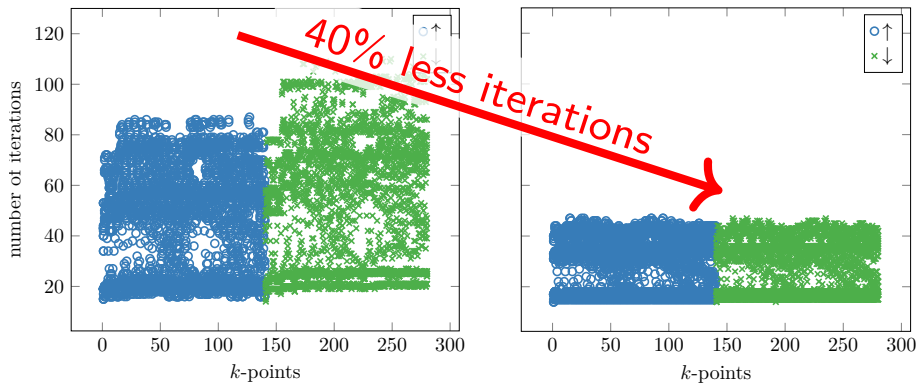


Fe_2MnAl Heusler alloy

k-point $[0.333, 0.0, 0.0]$



standard approach | Schur complement



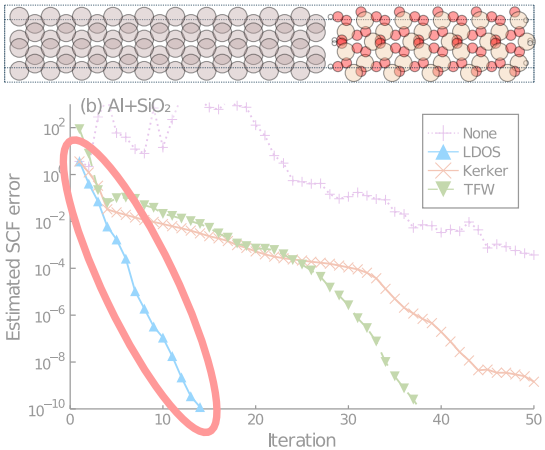
- Largest reduction in iterations near Fermi level ($n = 58$)
  (where gap is smallest)
- Overall 17% less iterations

$\Rightarrow$ Improvement comes for free
(extra bands needed during SCF)

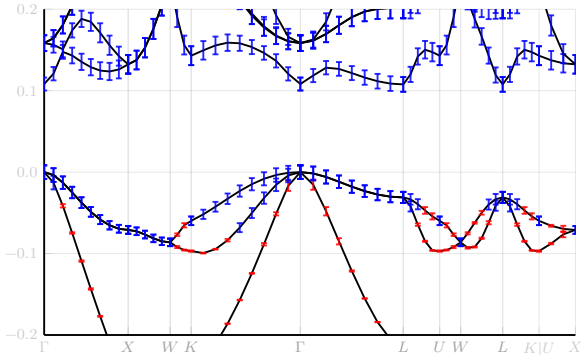
- Relevant materials class with unusual magnetic properties
- Translates to challenging numerical behaviour
- Schur-based approach tames CG
- ca. 40% less iterations

[1] E. Cancès, MFH, *et. al. Num. stability and efficiency of response property calculations in DFT* arXiv:2210.04512.

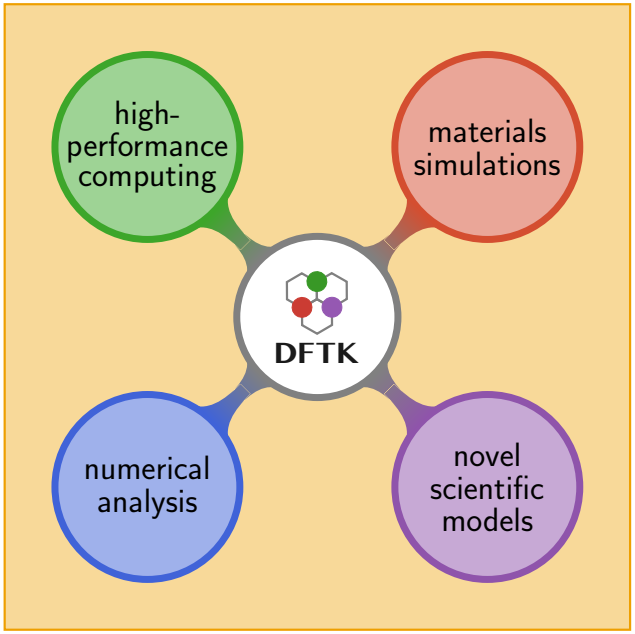
Black-box algorithms and robust error control for DFT



Self-adapting black-box DFT algorithms



Error control of DFT quantities

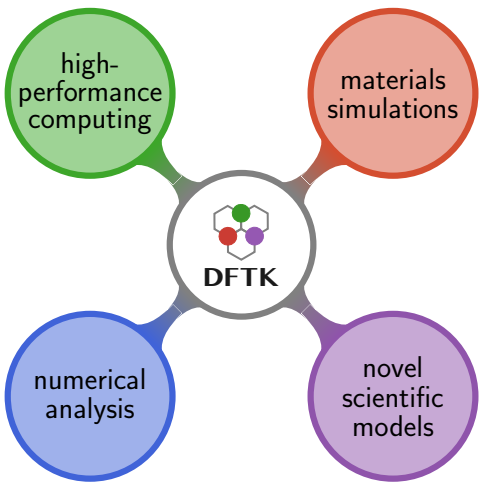


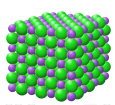
- Density-functional theory
- Local density of states preconditioner
- Adaptive damping

- Errors in DFT
- Model sensitivitites
- Algorithmic differentiation
- Robust response

- **DFTK** overview

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





$H\Psi = E\Psi$

- julia code for plane-wave DFT, started in 2019
- Fully composable with julia ecosystem:
  - Arbitrary precision (32bit, >64bit, . . .)
  - Algorithmic differentiation (AD)
  - HPC tools: GPU acceleration, MPI parallelisation
- Key tool in all presented research:
  - Allows restriction to relevant model problems,
  - *and* scale-up to application regime (1000 electrons)
  - Sizeable feature set in 7000 lines of code
  - Including some unique features (Self-adapting algorithms)

⇒ Build to enable multidisciplinary synergies

- Accessible high-productivity framework across domains:
  - Key code contributions by undegrads / PhD students
  - Initial AD support in 10 weeks (CS Bachelor)
  - Initial GPU support in 10 weeks (Physics Bachelor)
  - Relevant contributions from outside collab. circle

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)
)
```

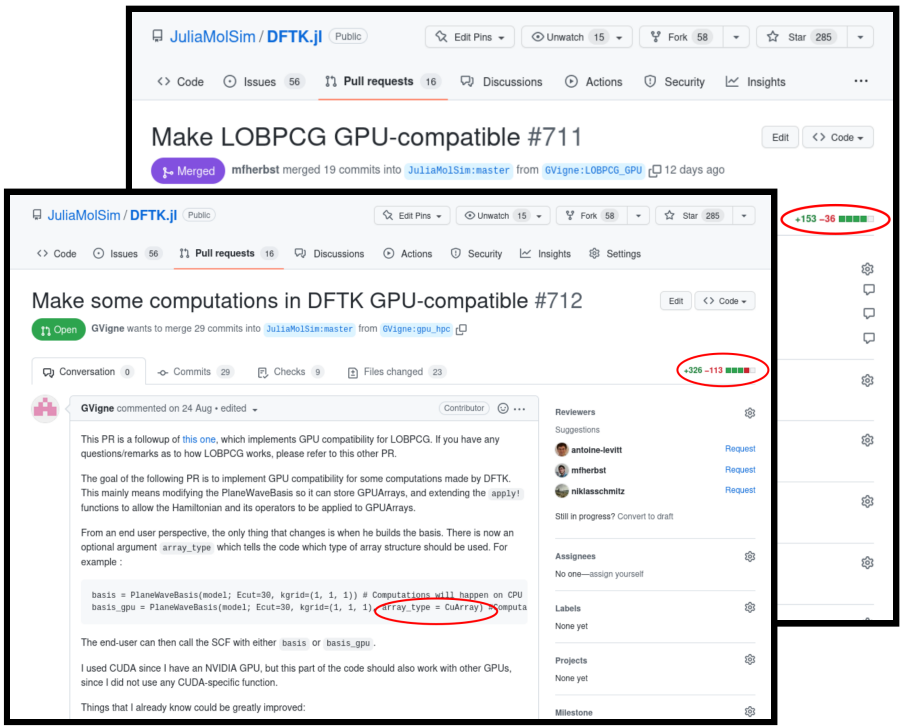
- Stress computation (Definition *vs.* julia code)[1]
- Post-processing step ⇒ Not performance critical
- Comparison of implementation complexity:
  - DFTK: 30 lines[1] (using forward-mode AD)
  - Quantum-Espresso: 1700 lines[2]
- Note: julia allows seamless composition of
  - Floating-point agnostic code for AD (slightly slower)
  - Fast code integrating with MPI, CUDA, . . .

⇒ 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

Preliminary GPU support in DFTK



- 10-week GSoC project
- < 500 lines changed
- Use julia's HPC abstractions to target all of CUDA, ROCm, oneAPI
- Painless installation and setup
- Collaboration with julia lab: CS, physics & maths

```
basis = PlaneWaveBasis(model; Ecut=30, kgrid=(1, 1, 1),
                       architecture=DFTK.GPU(CuArray))
```

DFTK: Bringing mathematical research to the applications

- Mathematical works with DFTK
  - Self-adapting black-box DFT methods[a,b]
  - Numerical analysis of DFT[c,d]
  - Practical error bounds[e,f]
- Exploring algorithmic differentiation:
  - "Automatic response": Phonons & higher-order properties
  - Data-enhanced DFT models
  - Full AD-able simulation pipeline: DFT, potentials, MD

  $\Rightarrow$ Uncertainty quantification all the way
- Part of growing julia materials modelling community
  - Common interfaces and data structures (e.g. AtomsBase)
- Outreach and teaching: DFTK summer school in 2022
  - United CS, maths, physics, chemistry, materials

Growing user base:

MIT, RWTH Aachen University, Carnegie Mellon University, École des Ponts ParisTech, Inria, UBFC Université Bourgogne Franche-Comté, TUM, VT, EPFL

[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] E. Cancès, MFH, *et. al. Num. stability and efficiency of response property calculations in DFT* arXiv:2210.04512.

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

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

Summary

https://michael-herbst.com/talks/2023.01.18_error_control_hu.pdf

- High-throughput screening
  - Main obstacle: Large number of parameters
  - Chosen empirically ⇒ Reliability limited
- Black-box strategies for damping & preconditioning
  - Build on combining mathematical and physical insight
  - Safeguard mechanism: Increase robustness for hard cases
  - Readily available in DFTK
- Response property calculations
  - Algorithmic differentiation: Many aspects left to explore
  - Prospects for code simplification
  - Routine computation of model sensitivities
  - First step towards data-enhanced DFT models
- DFTK: Multidisciplinary software development
  - julia-based framework for new DFT algorithms
  - In one code: Reduced problems and scale-up to realistic applications
  - High-productivity research framework

Acknowledgements

https://michael-herbst.com/talks/2023.01.18_error_control_hu.pdf

École des Ponts
Antoine Levitt
Eric Cancès

RWTH
Markus Towara

Mines Paris
Guillaume Vigne

MIT
Valentin Churavy
Katharine Fisher
Youssef Marzouk
Emmanuel Luján

TU Berlin
Niklas Schmitz

Uni Stuttgart
Benjamin Stamm

DFTK contributors

RWTH AACHEN UNIVERSITY

Inria

Summer of code

julia

MIT

Open PhD & PostDoc positions at EPFL, Switzerland

Possible topics include:

- Uncertainty quantification for DFT:
  Error in data-driven DFT models, pseudopotentials, propagation to properties and MD potentials
- Exploring algorithmic differentiation:
  Sensitivity analysis, higher-order properties
- Robust error-balancing simulations:
  Mixed precision, adaptive discretisation, model uncertainties, exploiting accelerators

- Interdisciplinary group: Math & Materials Science institutes
- Research embedded in high-throughput & ML activities
- Competitive Swiss salary & unique environment

- Contact me to discuss details

EPFL

Questions?

https://michael-herbst.com/talks/2023.01.18_error_control_hu.pdf

mfherbst

michael.herbst@epfl.ch

https://michael-herbst.com/blog

DFTK https://dftk.org

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

Applied and Computational Mathematics | RWTH AACHEN UNIVERSITY