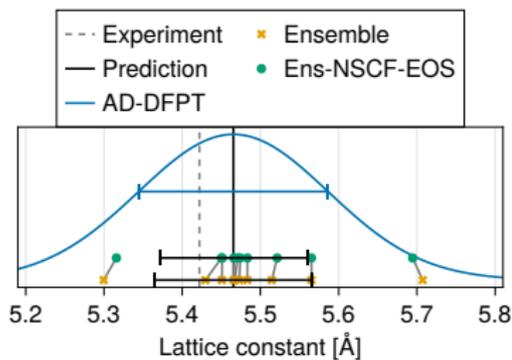
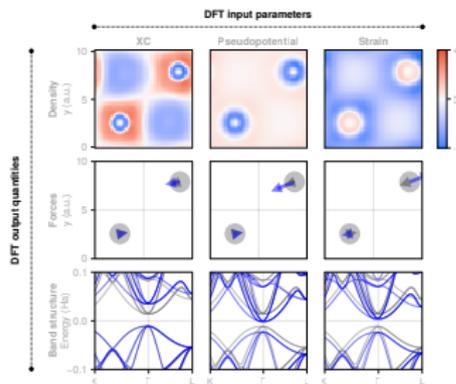


Algorithmic differentiation for plane-wave DFT:¹ materials design, error control and learning model parameters

Niklas F. Schmitz

Mathematics for Materials Modelling (matmat.org), EPFL

28th EuroAD Workshop @ CERN, Dec 9, 2025

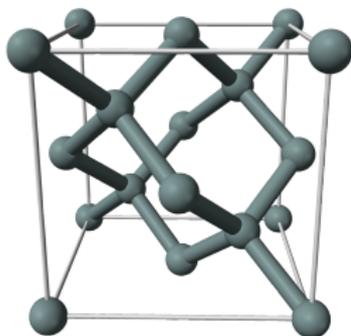


¹N. F. Schmitz, B. Ploumhans, M. F. Herbst. *npj Comput Mater*, (2025).
<https://doi.org/10.1038/s41524-025-01880-3>

Context: Ab-initio modeling & materials discovery

Applications: Novel batteries, solar cells, superconductors, ...

Structure at atomic scale \implies properties



Density-functional Theory (DFT):

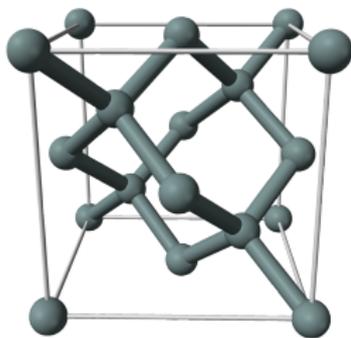
\implies Inputs: atomic species, positions, lattice

\implies Predicts electron density, total energy, forces, etc

Context: Ab-initio modeling & materials discovery

Applications: Novel batteries, solar cells, superconductors, ...

Structure at atomic scale \implies properties



Density-functional Theory (DFT):

\implies Inputs: atomic species, positions, lattice

\implies Predicts electron density, total energy, forces, etc

Workhorse method in computational materials science, eg.:

- ▶ High-throughput screening of materials databases
- ▶ Generates large-scale training data for materials foundation models

Parameter derivatives in DFT

- ▶ Physical responses \iff derivatives
 - ▶ Forces, stresses, phonons, dielectric response, ...
- ▶ Parameter optimization by gradients
 - ▶ inverse design of structure, ...
 - ▶ learning exchange-correlation, ...

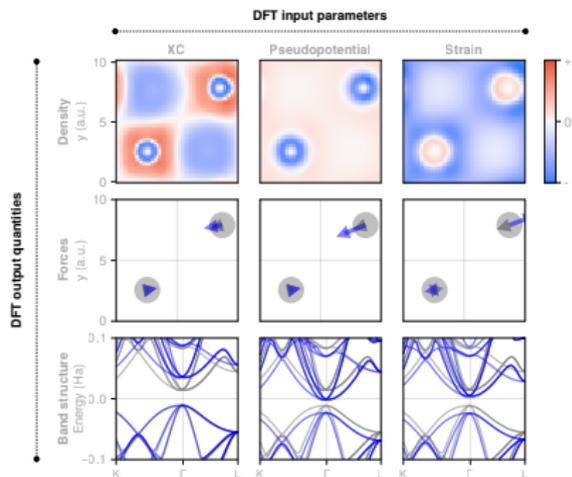


Figure: Computed with AD-DFPT in DFTK.jl (dftk.org)

Kohn-Sham Density Functional Theory (DFT)

Parameters θ (positions, lattice, model parameters,...)

Ground state: Find density matrix $P(\theta)$ minimizing energy $\mathcal{E}(\theta, P)$.

Self-consistent field (SCF) equations:

$$\begin{cases} H(\theta, P)\psi_n = \varepsilon_n\psi_n, & n = 1, \dots, N \\ P = \sum_{n=1}^N f(\varepsilon_n)\psi_n\psi_n^\dagger, \end{cases}$$

or compactly, writing f as a matrix function:

$$P = f(H(\theta, P)).$$

Plane-wave DFT: Large basis size N_b (avoid $\mathcal{O}(N_b^2)$ at all cost)

\implies matrix-free Hamiltonian (diagonal + low-rank + FFT convolutions)

\implies Iterative solvers working only on N orbitals are standard here
inner eigensolver (eg LOBPCG), outer fixed point (eg Anderson)

Density Functional Perturbation Theory (DFPT)

- ▶ Well-established method for DFT derivatives²
- ▶ Implicit differentiation of SCF equations at solution P :

$$P = f(H(\theta, P)) \implies \frac{\partial P}{\partial \theta} = (1 - \chi_0 K)^{-1} \chi_0 \frac{\partial H}{\partial \theta},$$

$\chi_0 := \frac{\partial f}{\partial H}$ the independent susceptibility of the density matrix, $K := \frac{\partial H}{\partial P}$ due to nonlinear Hartree and XC terms.

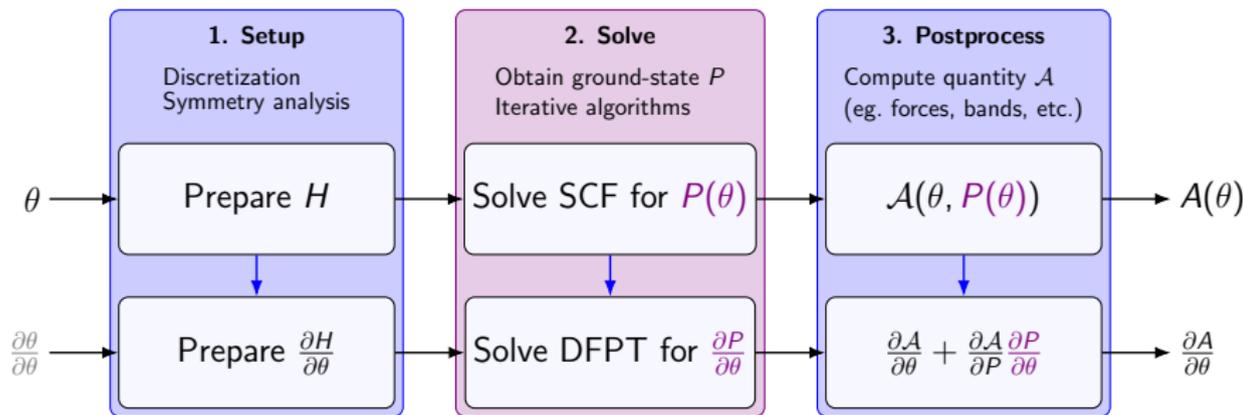
- ▶ Solve for $\frac{\partial P}{\partial \theta}$ (efficient matrix-free solvers on N orbitals)
(again nested: every χ_0 solves eigenvector PT by CG, outer solver is GMRES)
improved: Schur-complement preconditioning³, inexact GMRES⁴, ...

²Baroni, Gironcoli, Dal Corso, Giannozzi (2001). Phonons and related crystal properties from density-functional perturbation theory. Rev. Mod. Phys., 73, 515–562.

³Cancès, Herbst, Kemlin, Levitt, Stamm. Lett. in Math. Phys. 113, 21 (2023).

⁴Herbst, Sun. Efficient Krylov methods for lin. resp. in PW electronic structure calculations. <https://arxiv.org/abs/2505.02319>

End-to-end DFT derivatives with AD-DFPT⁷



- ▶ Setup and postprocessing handled by **general-purpose AD** system (ForwardDiff.jl)
- ▶ We **impose custom AD rule**: Converge SCF, then DFPT solver⁵⁶.

⁵Cancès, Herbst, Kemlin, Levitt, Stamm. Lett. in Math. Phys. 113, 21 (2023).

⁶Herbst, Sun. Efficient Krylov methods for lin. resp. in PW electronic structure calculations. <https://arxiv.org/abs/2505.02319>

⁷N. F. Schmitz, B. Ploumhans, M. F. Herbst. *npj Comput Mater*, (2025).
<https://doi.org/10.1038/s41524-025-01880-3>

Inverse design: Band gap by strain

a

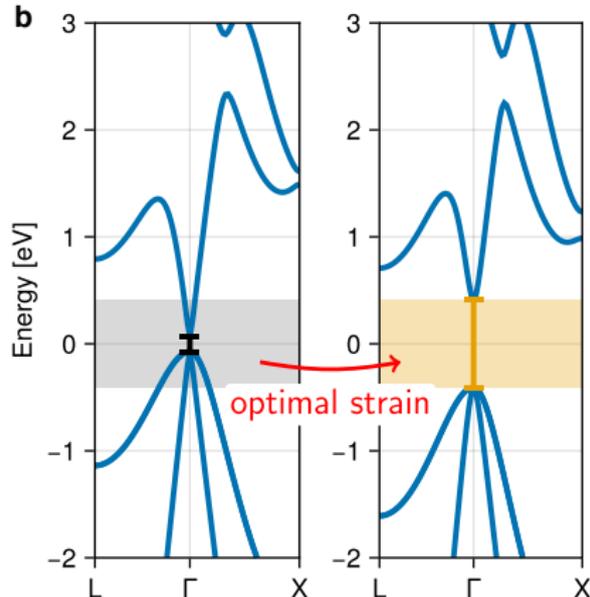
```
using DFTK, PseudoPotentialData, AtomsIO
using ForwardDiff, DifferentiationInterface, Optim

system = load_system("mp-2534-GaAs.cif")
pseudopotentials = PseudoFamily("dojo.nc.sr.pbe.v0_4_1.standard.upf")
model0 = model_DFT(system; functionals=PBE(), pseudopotentials,
    smearing=Smearing.Gaussian(), temperature=1e-3)

function strain_bandgap(η)
    model = Model(model0; lattice=(1 + η) * model0.lattice)
    basis = PlaneWaveBasis(model; Ecut=42, kgrid=(8, 8, 8))
    scfres = self_consistent_field(basis; tol=1e-6)
    eigenvalues_f = scfres.eigenvalues[1]
    ε_vbm = maximum(eigenvalues_f[eigenvalues_f .≤ scfres.εF])
    ε_cbm = minimum(eigenvalues_f[eigenvalues_f .> scfres.εF])
    ε_cbm - ε_vbm
end

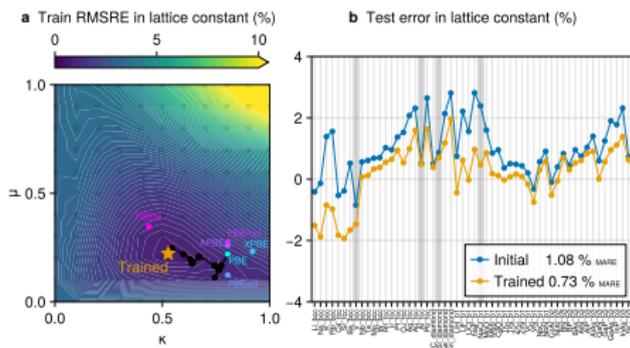
η0 = [0.0] # Initialize with zero strain (at equilibrium)
bandgap_target = 0.03 # Target band gap in [Ha]
bandgap_loss(η) = (bandgap_target - strain_bandgap(η[1]))^2
res = Optim.optimize(bandgap_loss, η0, BFGS(),
    Optim.Options(; iterations=5, x_abstol=1e-3);
    autodiff=AutoForwardDiff())
println("Optimized design strain η = ", res.minimizer)
```

b



- ▶ Easy: This is a full code example, just write a loss function with DFT
- ▶ `Optim.optimize` requests AD gradients, which triggers AD-DFPT

Learning exchange-correlation parameters⁸



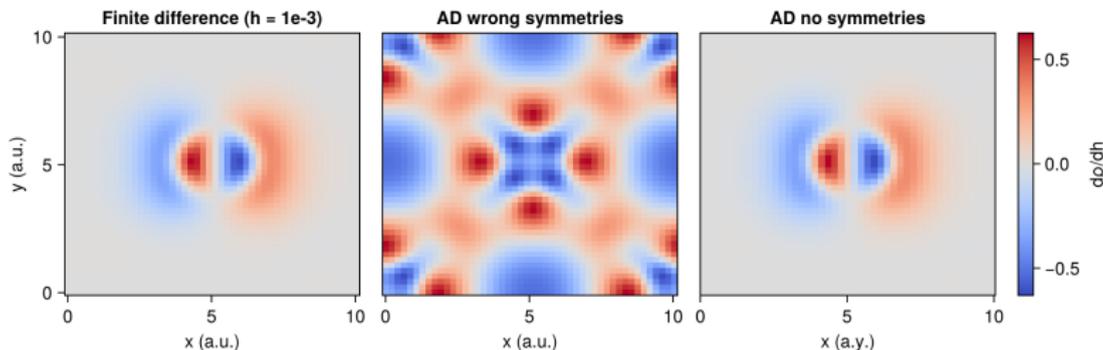
- ▶ Example: Training on experimental lattice constants
- ▶ Optimize a DFT functional (2 parameters θ) on $C = \text{Si, Al, V, NaCl}$

$$L_{\text{XC}}(\theta) = \frac{1}{N} \sum_{i=1}^N \left(\frac{a^*(C_i, \theta) - a_i^{\text{expt}}}{a_i^{\text{expt}}} \right)^2, \quad a^*(C, \theta) = \arg \min_a \min_P \mathcal{E}(C, a, \theta, P)$$

- ▶ Practical challenge to compute $\frac{\partial L_{\text{XC}}}{\partial \theta}$:
 - ▶ **Nested iterative methods** (eigensolver, SCF, lattice optimization)
 - ▶ Unusual derivative $\frac{\partial a^*}{\partial \theta} = - \left(\frac{\partial^2 E}{\partial a^2} \right)^{-1} \frac{\partial^2 E}{\partial \theta \partial a}$

⁸N. F. Schmitz, B. Ploumhans, M. F. Herbst. *npj Comput Mater*, (2025).
<https://doi.org/10.1038/s41524-025-01880-3>

Special: AD and symmetry-breaking perturbations

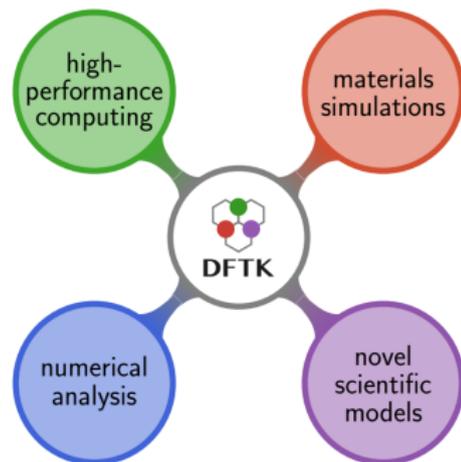


A symmetry-breaking density derivative $\frac{\partial \rho}{\partial \theta}$ computed in three ways.

- ▶ The problem: Our automated symmetry detection initially only considered the unperturbed crystal structure
- ▶ The fallback solution: Disable symmetries (expensive) or handle manually
- ▶ Better: determine stabilizer subgroup for perturbation automatically inside custom AD rule

Density-Functional ToolKit – <https://dftk.org>

- ▶ Implemented in Julia
- ▶ 10 000 lines, hackable & easy to extend (only 300 lines of custom ForwardDiff rules)
- ▶ HPC support: MPI, Nvidia & AMD GPUs
- ▶ Designed for cross-disciplinary research



Summary

- ▶ We introduced AD-DFPT
 - ▶ embeds DFPT into general-purpose AD system (ForwardDiff.jl)
 - ▶ use cases: Properties, Optimization, Uncertainty propagation, ...
 - ▶ Forward-mode scales with number of input parameters (just like DFPT)
- ▶ Future directions:
 - ▶ Losses and UQ on more properties (phonons, electric fields)
 - ▶ self-consistent ML-XC training on materials properties
 - ▶ extend DFT implementation generality (mGGA & nonlocal XC, ...)
 - ▶ extend to reverse-mode AD for gradients in high dimensions (Enzyme.jl+Reactant.jl?)

Thank you! Questions?

- ▶ N. F. Schmitz, B. Ploumhans, M. F. Herbst. *npj Comput Mater*, (2025).
<https://doi.org/10.1038/s41524-025-01880-3>
- ▶ Notebook:
https://showcases.matmat.org/2025/autodiff_dftk.html
- ▶ All examples: <https://github.com/niklasschmitz/ad-dfpt>
- ▶ Framework: DFTK.jl (<https://dftk.org>)

Appendix

Parameters in DFT

- ▶ Model (exchange-correlation, pseudopotentials, ...)
- ▶ Numerics (basis E_{cut} , \mathbf{k} -points, SCF tolerance, ...)

Tasks with gradients

- ▶ structure optimization, ...
- ▶ learning exchange-correlation, ...
- ▶ uncertainty propagation

Our new tool: Algorithmic
Differentiation for plane-wave DFT
(AD-DFPT)^a

^aN. F. Schmitz, B. Ploumhans, M. F. Herbst. *npj Comput Mater*, (2025). <https://doi.org/10.1038/s41524-025-01880-3>

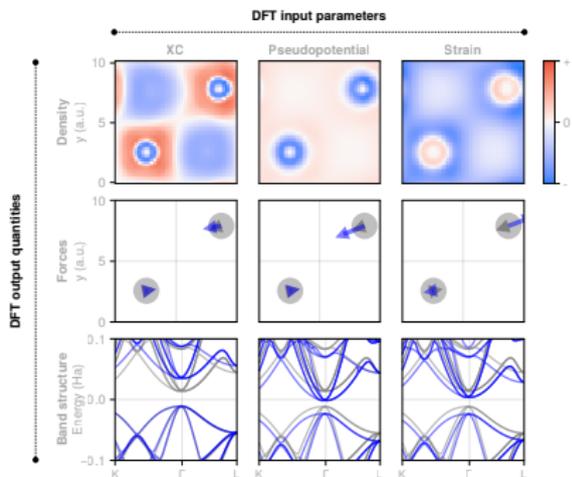
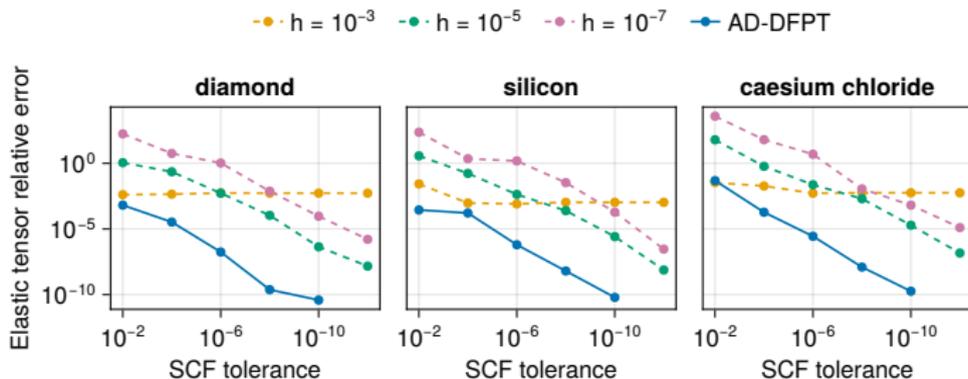


Figure: Derivatives from AD-DFPT in DFTK.jl (dftk.org)

Elasticity



- ▶ Accuracy better than finite differences, even at loose SCF tolerance
- ▶ Implementation simply by defining⁹ (strain η)
 - ▶ Stress $\sigma(\eta) = \frac{1}{V(\eta)} \frac{\partial \mathcal{E}}{\partial \eta}$, “Hellmann-Feynman Thm.”
 - ▶ Elastic tensor $C = \frac{\partial \sigma}{\partial \eta} \Big|_{\eta^*}$, AD-DFPT
- ▶ AD-DFPT combines precision and simplicity.

⁹https://docs.dftk.org/stable/examples/elastic_constants/

Forward uncertainty quantification¹⁰

- ▶ Given parameter uncertainty $\theta \sim \mathcal{N}(\theta_0, \Sigma)$, propagate to lattice constant $a^*(\theta)$

- ▶ Linearization around mean θ_0 :

$$a^*(\theta) \approx a^*(\theta_0) + J(\theta - \theta_0)$$

$\implies \mathcal{N}(a^*(\theta_0), J\Sigma J^T)$ with $J = \left. \frac{\partial a^*}{\partial \theta} \right|_{\theta_0}$ from AD-DFPT

- ▶ only 1 relaxation + 1 derivative, no N -fold ensemble required
- ▶ general

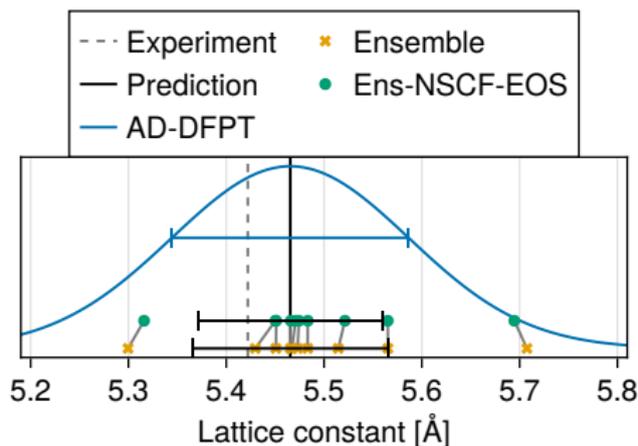


Figure: XC uncertainty (here BEEF^a) propagated to lattice constant on Si (AD-DFPT and Ensemble baselines)

^aMortensen, Kaasbjerg, Frederiksen, Nørskov, Sethna, Jacobsen. (2005). Phys. Rev. Lett., 95, 216401.

¹⁰N. F. Schmitz, B. Ploumhans, M. F. Herbst. *npj Comput Mater*, (2025). <https://doi.org/10.1038/s41524-025-01880-3>

Plane-wave discretization

Plane-waves are the Fourier modes of the periodic domain:

$$e_{\mathbf{G}}(\mathbf{r}) = |\Omega|^{-\frac{1}{2}} \exp(-i\mathbf{G} \cdot \mathbf{r}), \quad \mathbf{G} \in \mathcal{R}^*.$$

Variational approximation space:

$$X_{E_{\text{cut}}} := \text{Span} \left\{ e_{\mathbf{G}}, \quad \mathbf{G} \in \mathcal{R}^*, \quad \frac{1}{2}|\mathbf{G}|^2 \leq E_{\text{cut}} \right\}.$$

Strengths:

- ▶ systematic convergence using E_{cut}
- ▶ allows FFT-based methods

Weakness:

- ▶ lack of local refinement (\implies pseudopotentials)

Pseudopotentials



replacing inner electrons
with pseudopotential

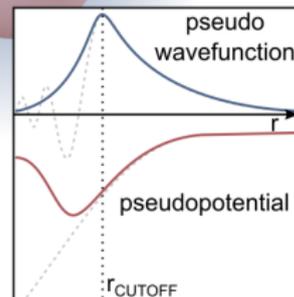
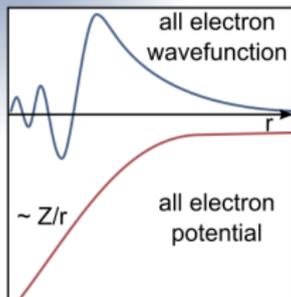
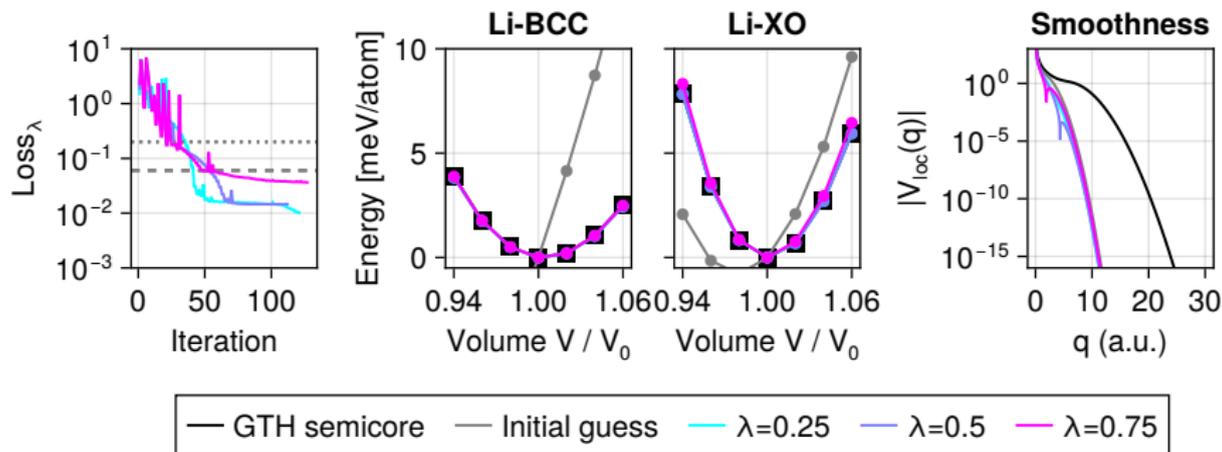


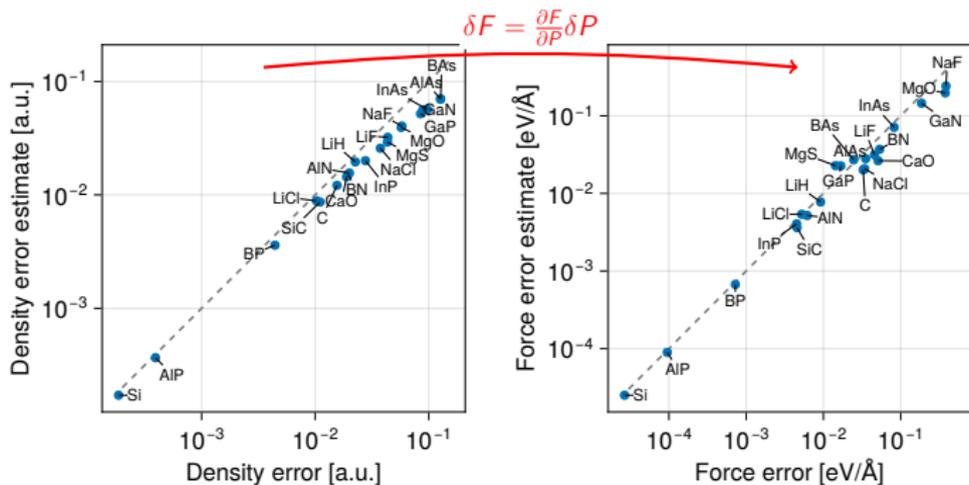
Figure: Edvin Fako, CC BY-SA 4.0, via Wikimedia Commons

Optimizing pseudopotentials¹¹



¹¹N. F. Schmitz, B. Ploumhans, M. F. Herbst. *npj Comput Mater*, (2025).
<https://doi.org/10.1038/s41524-025-01880-3>

Plane-wave error estimation¹³



Algorithmic differentiation (AD)

AD: automatic derivatives of computer programs by applying chain rule to primitive operations. $+$, $*$, matmul , ...

AD systems: Pytorch, JAX, Enzyme, ForwardDiff.jl, and many more

Two basic modes:

- ▶ Forward: sends input tangents v as $(x, v) \mapsto (f(x), Df(x)v)$
- ▶ Reverse: sends output tangents w to $Df(x)^\top w$
("backprop"; not yet in DFTK)

Key tool: **custom AD rules** to define new primitives

- ▶ implicit differentiation for numerical solvers \implies **AD-DFPT**
- ▶ foreign language calls (e.g. FFTW in C)
- ▶ symmetry detection algorithms