SCF preconditioning for mixed systems: A DFTK case study

Michael F. Herbst, Antoine Levitt

Matherials team, CERMICS, École des Ponts ParisTech

26th June 2020

MATHerials

Inria

École des Ponts ParisTech

→ https://michael-herbst.com/talks/2020.06.26_dftk_scf_emc2.pdf

Contents

Applications of DFT

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

- Interest in high-throughput screening
  - Drug design, catalysis, material science, ...
- As fast as possible, as accurate as necessary
- Reliable black-box workflows

Existing codes

- `https://www.vasp.at`
- `https://www.abinit.org`
- `http://www.castep.org`
- `https://wiki.fysik.dtu.dk/gpaw`
- `https://www.quantum-espresso.org`
- `https://crd-legacy.lbl.gov/~chao/KSSOLV`
- ...

- Since last year:  **DFTK**

⇒ But why another?

The essence of density-functional theory

$$\gamma_0 = \operatorname*{arg\,min}_{\gamma \in \mathcal{P}_N} \mathcal{E}_{\mathsf{DFT}}(\gamma)$$

- Energy functional (in an LDA-type model)

$$\mathcal{E}_{\mathsf{DFT}}(\gamma) = \mathrm{tr}_{L^2}\left(-\frac{1}{2}\Delta\gamma\right) + \int \rho_\gamma(\underline{\boldsymbol{r}}) V_{\mathsf{Nuc}}(\underline{\boldsymbol{r}})\,\mathrm{d}\underline{\boldsymbol{r}} + \frac{1}{2}\int \rho_\gamma(\underline{\boldsymbol{r}})\left[v_C(\underline{\boldsymbol{r}},\underline{\boldsymbol{r}}') + f_{\mathsf{xc},\rho_\gamma}(\underline{\boldsymbol{r}},\underline{\boldsymbol{r}}')\right]\rho_\gamma(\underline{\boldsymbol{r}}')\,\mathrm{d}\underline{\boldsymbol{r}}\,\mathrm{d}\underline{\boldsymbol{r}}'$$

- Density matrix $\gamma \in \mathcal{P}_N$
- Density $\rho_\gamma(\underline{\boldsymbol{r}}) = \gamma(\underline{\boldsymbol{r}},\underline{\boldsymbol{r}})$
- Coulomb kernel $v_C(\underline{\boldsymbol{r}},\underline{\boldsymbol{r}}') = 1/\left\|\underline{\boldsymbol{r}} - \underline{\boldsymbol{r}}'\right\|$
- Exchange-correlation kernel $f_{\mathsf{xc},\rho_\gamma}(\underline{\boldsymbol{r}},\underline{\boldsymbol{r}}')$

The self-consistent field procedure

- Euler-Lagrange equations (LDA):

$$\begin{cases} \gamma_0 = \mathbb{1}_{(-\infty,\varepsilon_F)}\left(-\frac{1}{2}\Delta + V_{\gamma_0}\right) & \text{with } \varepsilon_F \text{ s.t. } \gamma_0 \in \mathcal{P}_N \\ V_\gamma = V_{\text{Nuc}} + \int \left[v_C + f_{\text{xc},\rho_\gamma}\right] \rho_\gamma, \\ \rho_\gamma(\underline{\boldsymbol{r}}) = \gamma(\underline{\boldsymbol{r}}, \underline{\boldsymbol{r}}), \gamma \in \mathcal{P}_N \end{cases}$$

where

$$\mathbb{1}_{(-\infty,\varepsilon_F)}\left(\hat{\mathcal{F}}\right) = \sum_{\varepsilon_i < \varepsilon_F} |\psi_i\rangle\langle\psi_i| \qquad \text{with} \qquad \hat{\mathcal{F}}\psi_i = \varepsilon_i\psi_i$$

- Self-consistent field procedure:
  (1) Guess initial ρ_γ
  (2) Build Kohn-Sham Hamiltonian $-\frac{1}{2}\Delta + V_\gamma$
  (3) Diagonalise it to get new $\{\psi_i\}_i$
  (4) Build new ρ_γ, go to (2).

Plane-wave basis sets

- Periodic problems
- Eigenfunctions of Hamiltonian: Bloch waves $e^{\imath \underline{\boldsymbol{k}} \cdot \underline{\boldsymbol{r}}} u_{\underline{\boldsymbol{k}}}(\underline{\boldsymbol{r}})$
- Non-periodic part: $\underline{\boldsymbol{k}}$-point mesh in Brillouin zone
- Plane-wave basis sets for $u_{\underline{\boldsymbol{k}}}$

$$\varphi_{\underline{\boldsymbol{G}}}(\underline{\boldsymbol{r}}) = \frac{1}{\sqrt{|\Omega|}} e^{\imath \underline{\boldsymbol{G}} \cdot \underline{\boldsymbol{r}}}$$

- For given energy cutoff E_{cut}:

$$\left\{ \varphi_{\underline{\boldsymbol{G}}} \;\middle|\; \underline{\boldsymbol{G}} \in \mathcal{R}^*, \|\underline{\boldsymbol{G}} + \underline{\boldsymbol{k}}\|^2 \leq 2E_{\text{cut}} \right\}$$

- Notice: One cutoff, but different basis sets per $\underline{\boldsymbol{k}}$

Questions related to accuracy and reliability

- Accuracy
  - Which E_{cut}? A posteriori error estimates?
  - How to select $\underline{\boldsymbol{k}}$-point mesh? Convergence?
  - Which floating-point type? The same everywhere?
  - Error balancing procedures?
- Reliable self-consistent field procedures
  - What about spin, unusual functionals, disordered systems . . . ?
  - Which damping, mixing, eigensolver procedures, . . . ?
  - Can we get something black-box?

$\Rightarrow$ Mathematically-motivated questions

$\Rightarrow$ Direct impact to application domain

Demands for interdisciplinary software

- **Mathematicians:** Toy models and unphysical edge cases
- **Scientist:** Wants to focus on science, not numerics
- **High-performance person:** Exploit hardware specialities
- **Practitioner:** Reliable, black-box, high-level interface

- julia: Our language of choice
  - *Walks like Python, talks like Lisp, runs like FORTRAN*
  - Rich ecosystem (Optimisation, PDEs, stochastic processes, GPUs, Machine-Learning, Statistics, Linear Algebra . . .)
  - No two-language problem (high-level, compiled and hackable)

  $\Rightarrow$ Write code once, re-use for many back ends / machines . . .

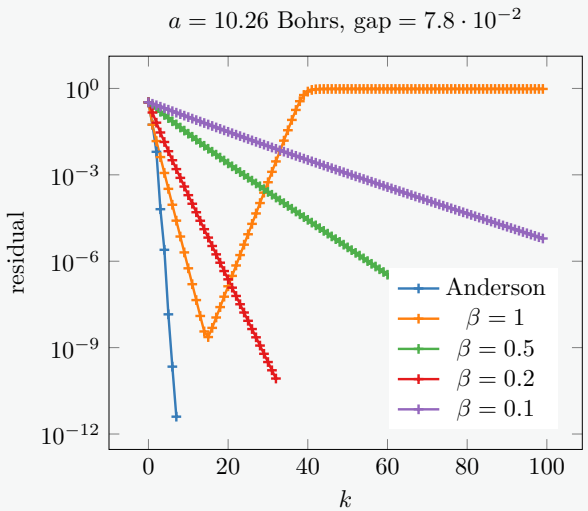
- `https://michael-herbst.com/learn-julia`

Our approach: DFTK, https://dftk.org

- 14 months of development, ≈ 5000 lines
- Sizeable feature list (see `https://docs.dftk.org`):
  - Ground state and a bit of response theory
  - Compose your model: Gross-Pitaevskii, analytic potentials ...
  - Multitude of SCF approaches
  - Multi-level threading
  - 1D / 2D / 3D systems
  - Arbitrary floating point type
  - Integration with materials-related python modules
- Performance: Within factor 2 of established codes
- Platform for multidisciplinary collaboration
- Documentation and examples: `https://docs.dftk.org`

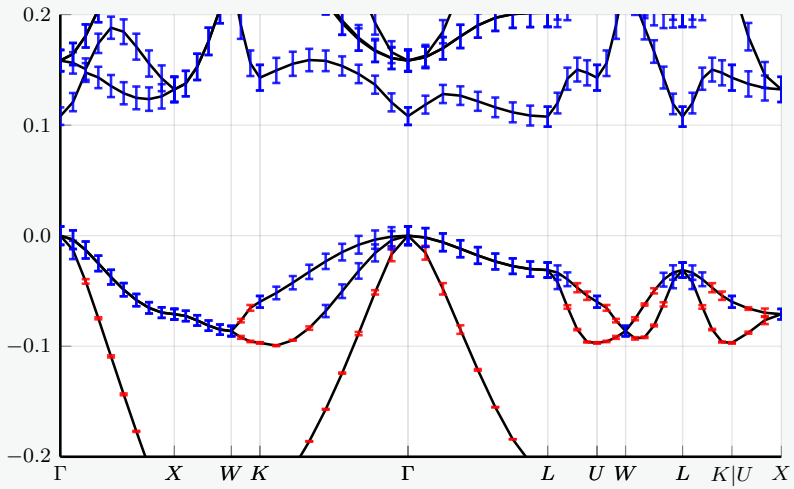
A few recent DFTK projects

Numerical analysis of SCF[1]



- SCF and direct minimisation
- Convergence wrt. spectral gap
- Numerical tests in DFTK

Error estimates for Kohn-Sham[2]



- A posteriori estimates for non-self-consistent Kohn-Sham
- Estimation of arithmetic error
- Elevated floating-point type
- Time to publication: 10 weeks

[1] E. Cancès, G. Kemlin, A. Levitt. arXiv 2004.09088 (2020)

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

DEMO

DEMO

A few DFTK examples

→ https://michael-herbst.com/talks/2020.06.26_dftk_scf_emc2_tutorials.ipynb

Contents

The SCF procedure as a fixed-point problem

- Recall the Euler-Lagrange equations

$$\begin{cases} \gamma_0 = \mathbb{1}_{(-\infty,\varepsilon_F)}\left(-\frac{1}{2}\Delta + V_{\gamma_0}\right) & \text{with } \varepsilon_F \text{ s.t. } \gamma_0 \in \mathcal{P}_N \\ V_\gamma = V_{\text{Nuc}} + \int \left[v_C + f_{\text{xc},\rho_\gamma}\right]\rho_\gamma, \\ \rho_\gamma(\underline{\boldsymbol{r}}) = \gamma(\underline{\boldsymbol{r}},\underline{\boldsymbol{r}}), \gamma \in \mathcal{P}_N \end{cases}$$

- Define the potential-to-density map F by

$$F(V)(\underline{\boldsymbol{r}}) = \left[\mathbb{1}_{(-\infty,\varepsilon_F)}\left(-\frac{1}{2}\Delta + V\right)\right](\underline{\boldsymbol{r}},\underline{\boldsymbol{r}})$$

and the density-to-potential map by

$$\mathcal{V}(\rho) = V_{\text{Nuc}} + \int \left[v_C + f_{\text{xc},\rho}\right]\rho$$

$\Rightarrow$ SCF solves $\rho = F(\mathcal{V}(\rho))$

The SCF Jacobian

- SCF solves $\rho = F(\mathcal{V}(\rho))$
- Corresponding Jacobian

$$J = 1 - \chi_0(v_C + f_{\text{xc},\rho})$$

χ_0: Independent-particle susceptibility (derivative of F)

- Physically: dielectric matrix ϵ

$\Rightarrow$ Convergence of SCF depends on dielectric properties

- Mixing schemes: Preconditioned quasi-Newton updates

$$\rho_{\text{next}} = \rho_{\text{in}} + P^{-1}\left(\rho_{\text{out}} - \rho_{\text{in}}\right)$$

where $P^{-1} \approx \epsilon^{-1}$ and $\rho_{\text{out}} = F(\mathcal{V}(\rho_{\text{in}}))$.

Typical mixing schemes

$$\rho_{\text{next}} = \rho_{\text{in}} + C\left(\rho_{\text{out}} - \rho_{\text{in}}\right)$$

- Insulators: Simple mixing $P^{-1} = 0 < \alpha \ll 1$
- Metals: Kerker mixing ($k_{\text{TF}} > 0$)

$$P^{-1} = \frac{-\alpha\Delta}{\frac{k_{\text{TF}}}{4\pi} - \Delta}, \qquad \hat{P}^{-1} = \frac{\alpha q^2}{q^2 + k_{\text{TF}}}$$

- Semiconductors: Resta mixing (relative permittivity ε_r)

$$\hat{P}^{-1} = \alpha \frac{k_F^2 - (1-\varepsilon_r)q^2}{\varepsilon_r k_F^2 - (1-\varepsilon_r)q^2}$$

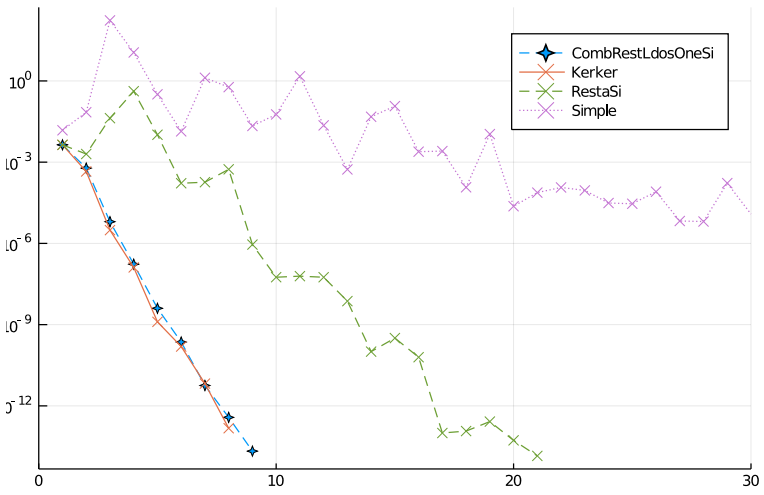
DEMO

DEMO

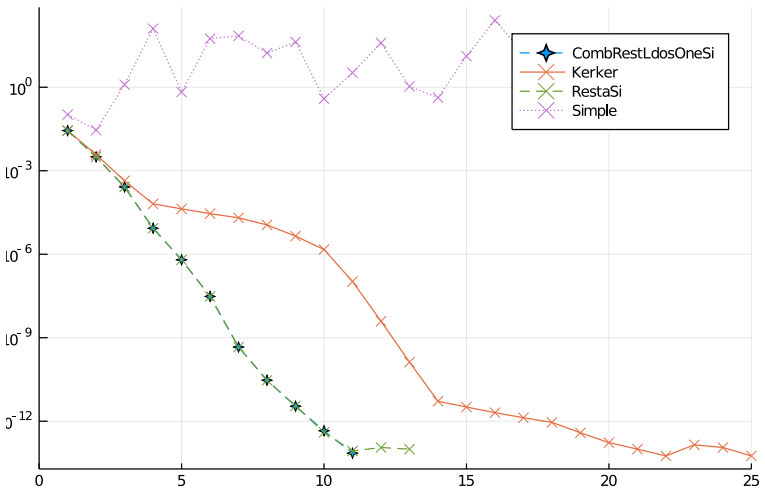
SCF preconditioners in 1D

→ `https://michael-herbst.com/talks/2020.06.26_dftk_scf_emc2_preconditioners.ipynb`

Bulk results in 3D



- Aluminium
- 10 randomised repeats
- $E_{\text{cut}} = 35$



- Silicon
- 10 randomised repeats
- $E_{\text{cut}} = 35$

Hybrid mixing

- What to do in mixed systems?
- Give up closed-form approximations for $\chi_0(\underline{\boldsymbol{r}}, \underline{\boldsymbol{r}}')$ in

$$\varepsilon^{-1} = \Big(1 - \chi_0(v_C + K_{\text{xc}})\Big)^{-1}$$

- Use *local* approximations and obtain ε^{-1} iteratively
- Based on a diagonal approximation of χ_0 one can argue ...

$$\int \chi_0(\underline{\boldsymbol{r}}, \underline{\boldsymbol{r}}') v(\underline{\boldsymbol{r}}') \,\mathrm{d}\underline{\boldsymbol{r}}' \approx -\mathrm{LDOS}(\varepsilon_F, \underline{\boldsymbol{r}}) v(\underline{\boldsymbol{r}}) + \frac{\mathrm{LDOS}(\varepsilon_F, \underline{\boldsymbol{r}})}{\int \mathrm{LDOS}(\varepsilon_F, \underline{\boldsymbol{r}}) \,\mathrm{d}\underline{\boldsymbol{r}}} \int \mathrm{LDOS}(\varepsilon_F, \underline{\boldsymbol{r}}') v(\underline{\boldsymbol{r}}') \,\mathrm{d}\underline{\boldsymbol{r}}'$$

  with $\mathrm{LDOS}(\underline{\boldsymbol{r}}) = \sum_n f'(\varepsilon_n) \left|\psi_n(\underline{\boldsymbol{r}})\right|^2$

- ... and do some large-scale testing

Hybrid mixing

- What to do in mixed systems?
- Give up closed-form approximations for $\chi_0(\underline{\boldsymbol{r}}, \underline{\boldsymbol{r}}')$ in

$$\varepsilon^{-1} = \Big(1 - \chi_0(v_C + K_{\text{xc}})\Big)^{-1}$$

- Use *local* approximations and obtain ε^{-1} iteratively
- Based on a diagonal approximation of χ_0 one can argue ...

$$\int \chi_0(\underline{\boldsymbol{r}}, \underline{\boldsymbol{r}}') v(\underline{\boldsymbol{r}}') \,\mathrm{d}\underline{\boldsymbol{r}}' \approx -\mathrm{LDOS}(\varepsilon_F, \underline{\boldsymbol{r}}) v(\underline{\boldsymbol{r}}) + \frac{\mathrm{LDOS}(\varepsilon_F, \underline{\boldsymbol{r}})}{\int \mathrm{LDOS}(\varepsilon_F, \underline{\boldsymbol{r}}) \,\mathrm{d}\underline{\boldsymbol{r}}} \int \mathrm{LDOS}(\varepsilon_F, \underline{\boldsymbol{r}}') v(\underline{\boldsymbol{r}}') \,\mathrm{d}\underline{\boldsymbol{r}}'$$

  with $\mathrm{LDOS}(\underline{\boldsymbol{r}}) = \sum_n f'(\varepsilon_n) \left|\psi_n(\underline{\boldsymbol{r}})\right|^2$

- ... and do some large-scale testing

Hybrid mixing (2)

- Unfortunately not quite as simple . . .

- For both insulators and semiconductors LDOS is basically zero

$\Rightarrow$ This scheme fails to distinguish them

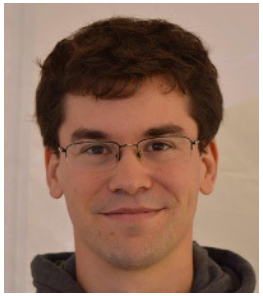
- . . . and we are not even talking about more involved cases

- Current approach: Add localised version of Resta mixing
- . . . and that's where we are now.

Summary and outlook

-  **DFTK**: Interdisciplinary software development
  - Feature-rich julia toolkit
  - Integration with established python packages
  - Rapid prototyping and toy problems
  - Scale-up to full-scale applications
- Hybrid mixing strategies
  - First step to a black-box preconditioner
  - Design motivated by understanding simple systems
  - Verification by considering production-grade tests

Acknowledgements


Antoine Levitt


Eric Cancès

Inria

École des Ponts ParisTech

SORBONNE UNIVERSITÉ

European Research Council
Established by the European Commission

Questions?

DFTK: `https://dftk.org`

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

mfherbst

`https://michael-herbst.com/blog`

`michael.herbst@inria.fr`