The Journal of Open Source Software

solid_dmft: gray-boxing DFT+DMFT materials simulations with TRIQS

Maximilian E. Merkel[1], **Alberto Carta**[1], **Sophie Beck**[2], **and Alexander Hampel**[2¶]

1 Materials Theory, ETH Zürich, Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland **2** Center for Computational Quantum Physics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Strongly correlated systems are a class of materials whose electronic structure is heavily influenced by the effect of electron-electron interactions. In these systems, an effective single-particle description may not capture the many-body effects accurately. Although density functional theory (DFT) plus dynamical mean-field theory (DMFT) has proven successful in describing strongly correlated electron systems for over two decades, it has only been very recently that ready-to-use software packages began to become available, with most scientific research carried out by self-written codes developed and used in research groups. Given the complexity of the method, there is also the question of whether users should implement the formalism themselves for each problem or whether ready-to-use black-box software, as is the case with many DFT software packages, is beneficial to the community.

The goal of solid_dmft is to find a middle ground, i.e., a *gray-box* tool as a ready-to-use implementation. This means that while the code contains all the functionality needed for many standard DMFT calculations, it is highly modular, based on open-source and community-developed software, and therefore can be easily adapted to specific applications and needs. Hence, this project is targeted towards researchers aiming to apply DMFT methods on top of DFT simulations to describe the physics of strongly correlated electron systems. While our approach allows one to fully perform these computations using standardized input flags without further need for elaborate coding, the final user can easily extend the functionalities by modifying the corresponding modules in the code.

The package is MPI-parallelized and written in Python 3, utilizing the publicly available TRIQS software library (Parcollet et al., 2015) and its available applications. The philosophy of the package is to increase reproducibility of DFT+DMFT calculations, provide clearer convergence metrics, and allow to run calculations for a large variety of systems without adapting the code manually, i.e., on a level similar to widely available DFT simulation packages.

Design Principles

solid_dmft uses the state-of-the-art implementations provided by the TRIQS library and its applications. This allows to easily run ab-initio calculations for strongly correlated materials, as well as implement and test new features of TRIQS and benchmark new solvers against existing ones. solid_dmft manages the calls of the necessary routines to run the DFT calculations, create the downfolded Hamiltonian, solve the resulting Hubbard-like Hamiltonian via DMFT, postprocess the data to calculate physical observables, and allow for charge-corrected feedback via charge self-consistency. The full DFT+DMFT cycle is presented in Figure 1.

The Journal of Open Source Software

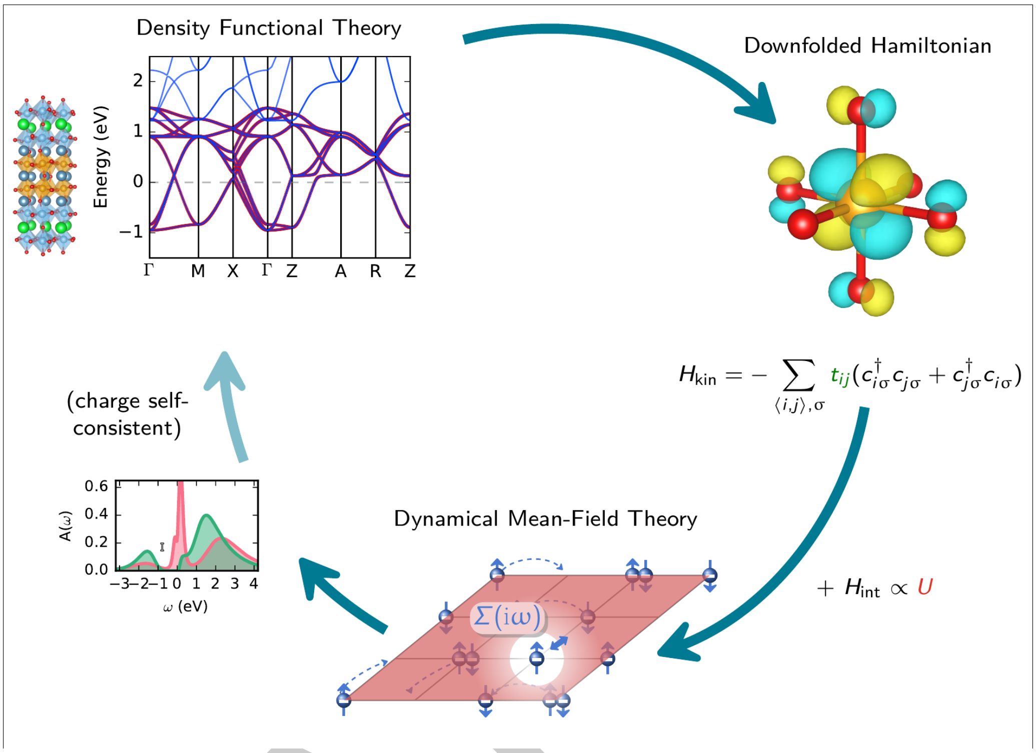

Figure 1: Fully charge self-consistent DFT+DMFT cycle. Starting from a DFT calculation (top left), a downfolded Hamiltonian and projector functions are created via optimized projections on a local basis set (top right). By adding a specified interaction Hamiltonian H_{int}, a full interacting electron problem is created, to be solved via the DMFT equations in TRIQS (bottom). After convergence in DMFT is reached, physical observables are calculated (bottom left). For fully charge self-consistent calculations, the DFT cycle is restarted with a DMFT-corrected charge density.

The code is designed to run on top of a DFT calculation or a tight-binding model, which provide the low-energy (downfolded) description of a periodic system. The DFT calculations can be performed with any code that is compatible with TRIQS/DFTTools (Aichhorn et al., 2016). The input for the DMFT calculation can be provided either as a Hamiltonian in reciprocal $\mathbf{k}$-space in a localized basis set or in terms of the overlap between the localized basis set and the Kohn-Sham wavefunctions (so-called projector functions) and their respective eigenvalues. The DFT output is converted by TRIQS/DFTTools into an HDF5 archive in a standardized structure to be used by solid_dmft.

The code follows the same modular philosophy as the TRIQS software package, relying on TRIQS functionalities to perform basic operations on Green functions. Each part of the simulation is split into separate stand-alone functions to limit statefulness to a minimum and allow for an easy extension of functionalities. The modularity of the program also allows to run, for example, the DMFT loop via a call of a single pure function with well-defined input and output, i.e., without running solid_dmft as a monolithic code. An abstracted `solver` class implements the various impurity solvers available in TRIQS. solid_dmft allows to seamlessly switch between impurity solvers with the change of a simple input flag and by adjusting the solver parameters. A fully charge self-consistent interface is implemented for Quantum ESPRESSO (Giannozzi et al., 2009) and the Vienna ab-initio simulation package (VASP) (Kresse & Furthmüller, 1996; Kresse & Hafner, 1993). solid_dmft also allows to perform inhomogeneous DMFT calculations, i.e., the treatment of multiple correlated and uncorrelated shells (impurity problems). Postprocessing scripts are available to perform analytic continuation of imaginary Green functions or self-energies, and to calculate spectral functions.

As of now, solid_dmft has been successfully used in various peer-reviewed research studies (Beck

& Georges, 2022; Hampel et al., 2019, 2020, 2021; Merkel & Ederer, 2021; Zhang et al., 2022). We provide releases matching those of the TRIQS library, as well as an updated documentation, tutorials, and a reference manual. Examples and benchmark calculations can be found in the tutorials section of the documentation. Furthermore, we utilize a continuous-integration workflow on GitHub to test every pull request and commit.

Statement of need

As of now, only few ready-to-use DFT+DMFT codes are available, all of them released rather recently. Most of these codes adopt a black-box approach, where the complexity of the DMFT part is abstracted away from the final user (as in EDMFT (Haule et al., 2010), Amulet (Poteryaev & others, n.d.) or the DMFT implementation in Abinit (Romero et al., 2020)) and therefore reduces the number of free parameters to tune. However, this approach may limit the flexibility of the implementation. solid_dmft is designed as a more modular, and open source implementation, similar to other software packages like DFTwDMFT (Singh et al., 2021) and DCORE (Shinaoka et al., 2021), and provides a flagship implementation of the TRIQS functionality. The benefits of this approach are twofold: on the one hand, developers of the TRIQS ecosystem are able to benchmark their applications in a well-tested framework. On the other hand, users can benefit from a standardized input-output structure compatible with the TRIQS framework, fundamentally increasing robustness and reproducibility. solid_dmft is developed in the spirit of a community code and supports external contributions that advance the capabilities of the software.

Acknowledgements

This research was supported by ETH Zurich and the NCCR MARVEL, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 182892). The Flatiron Institute is a division of the Simons Foundation.

References

Aichhorn, M., Pourovskii, L., Seth, P., Vildosola, V., Zingl, M., Peil, O., Deng, X., Mravlje, J., Kraberger, G. J., Martins, C., Ferrero, M., & Parcollet, O. (2016). TRIQS/DFTTools: A TRIQS application for ab initio calculations of correlated materials. *Computer Physics Communications*, *204*, 200–208. https://doi.org/doi:10.1016/j.cpc.2016.03.014

Beck, H., S., & Georges, A. (2022). Charge self-consistent electronic structure calculations with dynamical mean-field theory using quantum ESPRESSO, wannier 90 and TRIQS. *Journal of Physics: Condensed Matter*, *34*, 235601. https://doi.org/10.1088/1361-648x/ac5d1c

Giannozzi, P., Baroni, S., Bonini, N., Calandra, M., Car, R., Cavazzoni, C., Ceresoli, D., Chiarotti, G. L., Cococcioni, M., Dabo, I., Corso, A. D., Gironcoli, S. de, Fabris, S., Fratesi, G., Gebauer, R., Gerstmann, U., Gougoussis, C., Kokalj, A., Lazzeri, M., … Wentzcovitch, R. (2009). Quantum ESPRESSO: A modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter*, *21*, 395502. https://doi.org/10.1088/0953-8984/21/39/395502

Hampel, A., Beck, S., & Ederer, C. (2020). Effect of charge self-consistency in $\mathrm{DFT}+\mathrm{DMFT}$ calculations for complex transition metal oxides. *Phys. Rev. Research*, *2*, 033088. https://doi.org/10.1103/PhysRevResearch.2.033088

Hampel, A., Lee-Hand, J., Georges, A., & Dreyer, C. E. (2021). Correlation-induced octahedral rotations in SrMoO_3. *Phys. Rev. B*, *104*, 035102. https://doi.org/10.1103/PhysRevB.104.035102

Hampel, A., Liu, P., Franchini, C., & Ederer, C. (2019). Energetics of the coupled electronic–structural transition in the rare-earth nickelates. *Npj Quantum Materials*, *4*, 5. https://doi.org/10.1038/s41535-019-0145-4

Haule, K., Yee, C.-H., & Kim, K. (2010). Dynamical mean-field theory within the full-potential methods: Electronic structure of $\mathrm{CeIrIn_5}$, $\mathrm{CeCoIn_5}$, and $\mathrm{CeRhIn_5}$. *Phys. Rev. B*, *81*, 195107. https://doi.org/10.1103/PhysRevB.81.195107

Kresse, G., & Furthmüller, J. (1996). Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical Review B*, *54*, 11169–11186. https://doi.org/10.1103/PhysRevB.54.11169

Kresse, G., & Hafner, J. (1993). Ab initio molecular dynamics for liquid metals. *Physical Review B*, *47*, 558–561. https://doi.org/10.1103/PhysRevB.47.558

Merkel, M. E., & Ederer, C. (2021). Charge disproportionation and Hund's insulating behavior in a five-orbital Hubbard model applicable to d^4 perovskites. *Physical Review B*, *104*, 165135. https://doi.org/10.1103/PhysRevB.104.165135

Parcollet, O., Ferrero, M., Ayral, T., Hafermann, H., Krivenko, I., Messio, L., & Seth, P. (2015). TRIQS: A toolbox for research on interacting quantum systems. *Computer Physics Communications*, *196*(Supplement C), 398–415. https://doi.org/https://doi.org/10.1016/j.cpc.2015.04.023

Poteryaev, A., & others. (n.d.). http://amulet-code.org.

Romero, A. H., Allan, D. C., Amadon, B., Antonius, G., Applencourt, T., Baguet, L., Bieder, J., Bottin, F., Bouchet, J., Bousquet, E., Bruneval, F., Brunin, G., Caliste, D., Côté, M., Denier, J., Dreyer, C., Ghosez, P., Giantomassi, M., Gillet, Y., … Gonze, X. (2020). ABINIT: Overview and focus on selected capabilities. *The Journal of Chemical Physics*, *152*(12), 124102. https://doi.org/10.1063/1.5144261

Shinaoka, H., Otsuki, J., Kawamura, M., Takemori, N., & Yoshimi, K. (2021). DCore: Integrated DMFT software for correlated electrons. *SciPost Phys.*, *10*, 117. https://doi.org/10.21468/SciPostPhys.10.5.117

Singh, V., Herath, U., Wah, B., Liao, X., Romero, A. H., & Park, H. (2021). DMFTwDFT: An open-source code combining dynamical mean field theory with various density functional theory packages. *Computer Physics Communications*, *261*, 107778. https://doi.org/https://doi.org/10.1016/j.cpc.2020.107778

Zhang, R., Merkel, M. E., Beck, S., & Ederer, C. (2022). *Training biases in machine learning for the analytic continuation of quantum many-body Green's functions*. arXiv. https://doi.org/10.48550/arXiv.2206.07493